



Программный комплекс Систэм Платформ

SePlatform.HMI.CommonLib 2.0

Руководство пользователя



© ООО «СИСТЭМ СОФТ», 2022-2024. Все права защищены.

Авторские права на данный документ принадлежат ООО «СИСТЭМ СОФТ». Копирование, перепечатка и публикация любой части или всего документа не допускается без письменного разрешения правообладателя.

Содержание

1. Руководство пользователя	6
1.1. О продукте	6
1.2. Подготовка к работе	6
1.3. Включение расширения в проект автоматизации	7
1.4. Примеры использования компонентов расширения	9
1.4.1. Выбор цвета на панели инструментов из палитры или из меню	9
2. Справочное руководство	12
2.1. Типы	12
2.1.1. Тема оформления	12
2.1.1.1. Свойства	13
2.1.1.2. Функции	15
2.1.2. Конвертер значений	16
2.1.2.1. Функции	17
2.1.3. Утилиты для работы со строками	21
2.1.3.1. Функции	21
2.1.4. Утилиты для работы с окном	22
2.1.4.1. Свойства	22
2.1.4.2. Функции	22
2.1.5. Информация о продукте	24
2.1.5.1. Свойства	24
2.1.5.2. Функции	25
2.1.6. Буфер аргументов события	25
2.1.6.1. Свойства	27
2.1.6.2. Функции	28
2.1.7. Обработчик события	28
2.1.7.1. Свойства	29
2.1.7.2. Функции	29
2.1.7.3. События	29
2.1.8. Текстовые шаблоны	30
2.1.8.1. Свойства	30
2.1.8.2. Функции	30
2.1.9. Редакторы	32
2.1.9.1. Редактор целого числа	32
Свойства	33
2.1.9.2. Редактор даты	33
Свойства	34
2.1.9.3. Редактор времени	34
Свойства	35
2.1.9.4. Редактор интервала времени	35
Свойства	36
2.1.9.5. Горизонтальный сплиттер	36
Свойства	38
2.1.9.6. Вертикальный сплиттер	39
Свойства	40
2.1.10. Меню	41
2.1.10.1. Элемент меню	41
Свойства	41
2.1.10.2. Разделитель элементов меню	42
2.1.10.3. Результат выбора меню	43
Свойства	43
2.1.11. Панель инструментов	43
2.1.11.1. Кнопка панели инструментов	44

Свойства	45
2.1.11.2. Кнопка выпадающего списка	46
Свойства	46
2.1.11.3. Разделитель панели инструментов	47
Свойства	47
2.1.11.4. Панель поиска	48
Свойства	48
2.1.11.5. Тема оформления панели инструментов	49
Свойства	49
2.1.12. Строка статуса	51
2.1.12.1. Кнопка авторизации	51
Свойства	51
2.1.12.2. Кнопка-индикатор журнала сообщений	51
Свойства	52
2.1.12.3. Индикатор подключения к серверу	52
Свойства	52
2.1.12.4. Кнопка смены раскладки клавиатуры	53
2.1.12.5. Кнопка вызова экранной клавиатуры	53
Свойства	54
2.1.12.6. Параметры вызова экранной клавиатуры	54
Свойства	54
2.1.13. Диалоги	55
2.1.13.1. Результат выбора папки	55
Свойства	55
2.1.13.2. Результат выбора цвета	55
Свойства	55
2.1.13.3. Результат выбора параметров подключения	56
Свойства	56
2.1.13.4. Результат выбора даты и времени	57
Свойства	57
2.1.13.5. Результат ввода значения	57
Свойства	57
2.1.13.6. Результат редактирования колонтитула	59
Свойства	59
2.1.13.7. Результат выбора файла	59
Свойства	59
2.1.13.8. Результат диалога авторизации	60
Свойства	60
2.1.13.9. Список файлов	60
Свойства	61
2.1.14. Дерево сигналов	63
2.1.14.1. Свойства	63
2.1.14.2. Функции	64
2.1.15. Журнал сообщений	66
2.1.15.1. Свойства	66
2.1.15.2. Функции	67
2.2. Экранные формы	70
2.2.1. Меню	70
2.2.1.1. Свойства	70
2.2.2. Календарь	73
2.2.2.1. Свойства	74
2.2.3. Сообщение	74
2.2.3.1. Свойства	75
2.2.4. Диалог ввода значения	77
2.2.4.1. Свойства	77
2.2.5. Диалог выбора цвета	78

2.2.5.1. Свойства	79
2.2.6. Диалог подключения к серверу	80
2.2.6.1. Свойства	80
2.2.7. Редактор колонтитула	82
2.2.7.1. Свойства	82
2.2.8. Файлы	83
2.2.8.1. Диалог открытия файла	83
Свойства	84
2.2.8.2. Диалог сохранения файла	88
Свойства	88
2.2.8.3. Диалог выбора папки	93
Свойства	93
2.2.9. Безопасность	94
2.2.9.1. Диалог авторизации пользователя	94
Свойства	95
2.2.9.2. Диалог смены пароля	96
Свойства	96
2.2.9.3. Диалог ввода учетных данных	97
Свойства	97
2.3. Общие свойства компонентов библиотеки	97
2.3.1. Тема оформления (pTheme)	97
История изменений	99
2.0	99
2.0.1	99
2.0.2	100
Изменения документации	100
Редакция 1	100
Редакция 2	101
Редакция 3	101
1.4	101
1.4.1	102
1.4.2	102
1.4.3	102
Изменения документации	102
Редакция 1	102
Редакция 2	103
1.3	103
1.3.1	103
Изменения документации	104
Редакция 1	104
Редакция 2	104
1.2	104
1.2.2	105
Изменения документации	105
Редакция 1	105
Редакция 2	105
1.1	105
1.1.1	105
Изменения документации	106
Редакция 2	106

1. Руководство пользователя

1.1. О продукте

SePlatform.HMI.CommonLib - это расширение среды разработки и исполнения SePlatform.HMI.

Расширение представляет собой библиотеку компонентов, которые можно использовать в своих проектах автоматизации:

- диалоговые окна;
- контекстное меню;
- календарь;
- файловый менеджер;
- конвертер значений;
- дерево сигналов;
- журнал сообщений;
- компоненты безопасности;
- различные кнопки и индикаторы.

Внешний вид и функциональность компонентов можно настроить с помощью свойств и функций. Подробнее назначение, свойства и функции каждого компонента описаны в Справочном руководстве.

Расширение также используется для работы некоторых компонентов Систем Платформ. Поэтому его часто необходимо устанавливать и встраивать в проекты автоматизации, разработанные в SePlatform.HMI, наряду с другими расширениями.

1.2. Подготовка к работе

Требования к окружению

Для использования средств SePlatform.HMI.CommonLib должны быть установлены:

- среда разработки и исполнения SePlatform.HMI;
- подсистема безопасности SePlatform.Security;
- расширения SePlatform.HMI.Tables и SePlatform.HMI.Security среды SePlatform.HMI.

Если работаете с веб-версией проекта автоматизации, установите дополнительно:

- SePlatform.HMI.WebViewer - для использования проектов SePlatform.HMI в веб-интерфейсе;
- SePlatform.HMI.Security.WebViewer - для работы компонентов SePlatform.HMI.Security в веб-интерфейсе;
- SePlatform.HMI.Tables.WebViewer - для работы компонентов SePlatform.HMI.Tables в веб-интерфейсе.

Установка и удаление на ОС Windows

Для установки, удаления или восстановления SePlatform.HMI.CommonLib используйте мастер. Запустите установочный файл `seplatform.hmi.commonlib-<lng>-<version>.<arch>.msi`.

**ПРИМЕЧАНИЕ**

В названии файла <lng> - это язык продукта, <version> - версия продукта, <arch> - целевая процессорная архитектура.

Следуйте инструкциям мастера.

Расширение устанавливается в папку C:\Program Files\SePlatform\SePlatform.HMI.Extensions\CommonLib или C:\Program Files (x86)\SePlatform\SePlatform.HMI.Extensions\CommonLib.

Установка и удаление на ОС Linux

Для установки SePlatform.HMI.CommonLib вызовите пакет с командой на установку:

➤ rpm-пакет:

```
sudo rpm -i seplatform.hmi.commonlib-<lng>-<version>.rpm
```

➤ deb-пакет:

```
sudo dpkg -i seplatform.hmi.commonlib-<lng>-<version>.deb
```

**ПРИМЕЧАНИЕ**

В названии файла <lng> - это язык продукта, а <version> - версия продукта.

Расширение устанавливается в папку /opt/SePlatform/SePlatform.HMI.Extensions/CommonLib.

Для удаления SePlatform.HMI.CommonLib вызовите пакет с командой на удаление:

➤ rpm-пакет:

```
sudo rpm -e seplatform.hmi.commonlib
```

➤ deb-пакет:

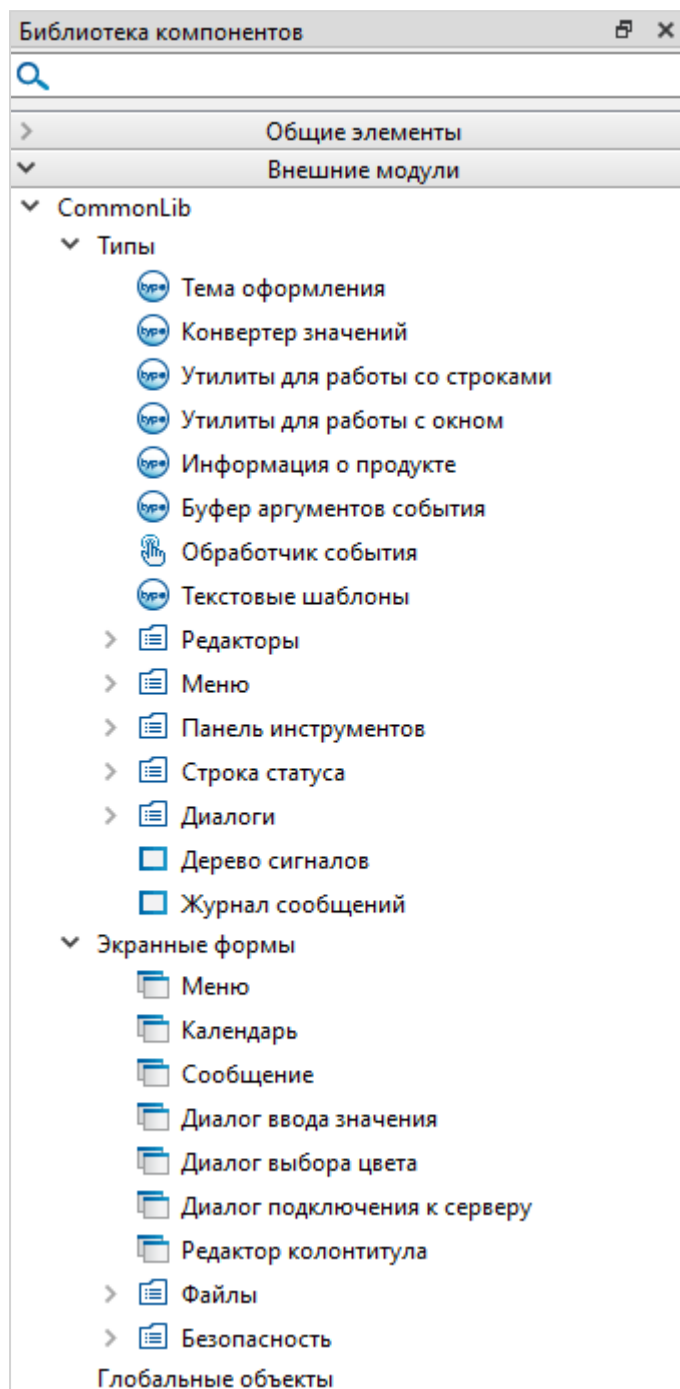
```
sudo dpkg -r seplatform.hmi.commonlib
```

1.3. Включение расширения в проект автоматизации

Расширение SePlatform.HMI.CommonLib выполнено в виде проекта SePlatform.HMI. Чтобы использовать компоненты расширения, необходимо подключить проект расширения в свой проект SePlatform.HMI в виде внешнего модуля.

1. Чтобы подключать проекты расширений в виде внешних модулей, создайте в папке своего проекта папку с именем externals. Если к проекту уже подключены другие модули, папка externals может уже существовать.
2. В созданную папку скопируйте папку расширения CommonLib, расположенную в:
 - C:\Program Files\SePlatform\SePlatform.HMI.Extensions или C:\Program Files (x86)\SePlatform\SePlatform.HMI.Extensions (ОС Windows);
 - /opt/SePlatform/SePlatform.HMI.Extensions (ОС Linux).

3. Откройте свой проект в Дизайнер SePlatform.HMI. В меню **Проект** на панели инструментов выберите пункт **Обновить ссылки на внешние юниты**. В результате в библиотеке компонентов появится юнит внешнего модуля **CommonLib**.



Чтобы приступить к работе, перетяните компонент на форму или вызовите из обработчика какого-либо события. В проект можно добавить столько экземпляров одного типа, сколько необходимо. Разные экземпляры одного типа можно использовать под разные цели.

Обновление расширения

Если SePlatform.HMI.CommonLib уже встроен в проект в виде внешнего модуля, то при установке новой версии необходимо обновить файлы модуля в папке externals проекта. Для этого:

1. Установите нужную версию SePlatform.HMI.CommonLib.
2. Замените существующую папку CommonLib на папку CommonLib, расположенную в папке установки расширения:
 - C:\Program Files\SePlatform\SePlatform.HMI.Extensions или C:\Program Files (x86)\SePlatform\SePlatform.HMI.Extensions (OC Windows);
 - /opt/SePlatform/SePlatform.HMI.Extensions (OC Linux).
3. Откройте свой проект в Дизайнер SePlatform.HMI. В меню **Проект** на панели инструментов выберите **Обновить ссылки на внешние юниты**. В результате внешние модули будут обновлены. Экземпляры компонентов SePlatform.HMI.CommonLib, уже добавленные в проект, обновятся автоматически.

1.4. Примеры использования компонентов расширения

1.4.1. Выбор цвета на панели инструментов из палитры или из меню



ПРИМЕЧАНИЕ

В разделе продемонстрировано применение компонентов [2.1.11.1. Кнопка панели инструментов \(стр. 44\)](#), [2.1.11.2. Кнопка выпадающего списка \(стр. 46\)](#), [2.2.5. Диалог выбора цвета \(стр. 78\)](#) и [2.2.1. Меню \(стр. 70\)](#). Для этого в SePlatform.HMI реализован проект-пример, который можно скачать для ознакомления.

Предположим, необходимо создать панель инструментов, предназначенную для выбора цвета из палитры или из выпадающего меню. Решим эту задачу, используя комбинацию компонентов [2.1.11.1. Кнопка панели инструментов \(стр. 44\)](#), [2.1.11.2. Кнопка выпадающего списка \(стр. 46\)](#), [2.2.5. Диалог выбора цвета \(стр. 78\)](#) и [2.2.1. Меню \(стр. 70\)](#):

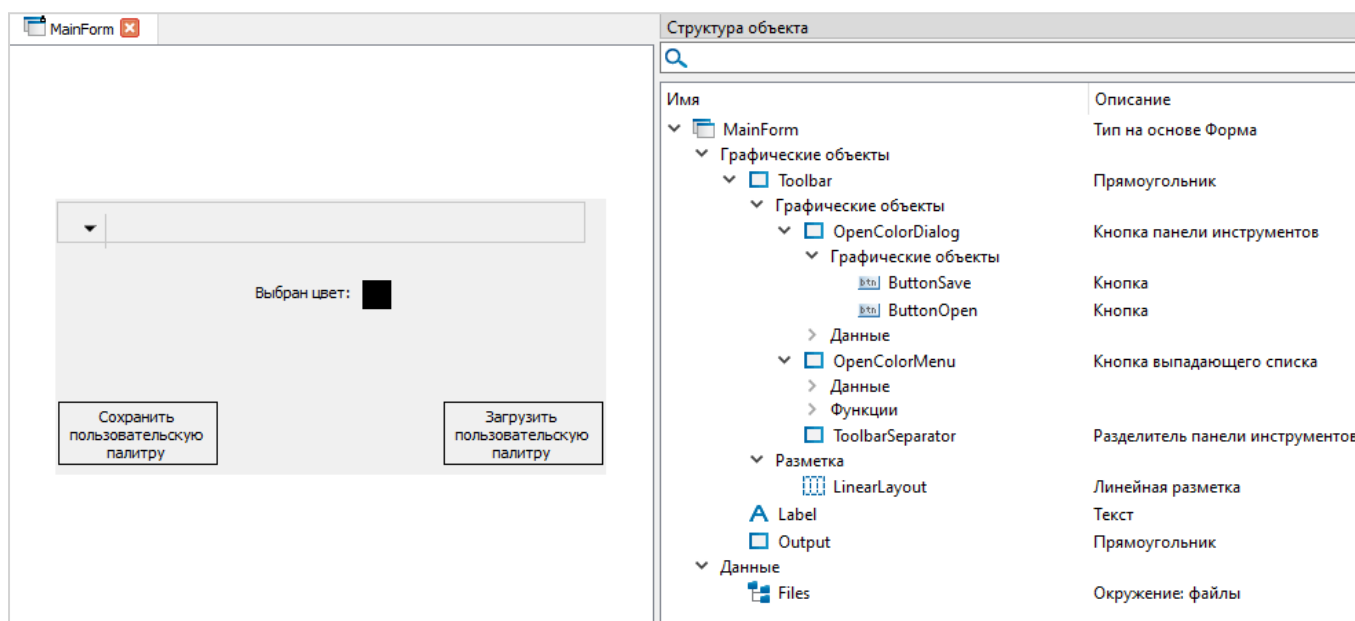
- По нажатию на кнопку типа [2.1.11.1. Кнопка панели инструментов \(стр. 44\)](#) открывается палитра [2.2.5. Диалог выбора цвета \(стр. 78\)](#). Пусть при этом необходимо иметь возможность использовать пользовательскую палитру цветов.
- По нажатию на кнопку типа [2.1.11.2. Кнопка выпадающего списка \(стр. 46\)](#) открывается [2.2.1. Меню \(стр. 70\)](#), где можно выбрать цвет из предложенных.

Решение реализовано в проекте-примере. Скачайте проект-пример и откройте в Дизайнер SePlatform.HMI для наглядности.

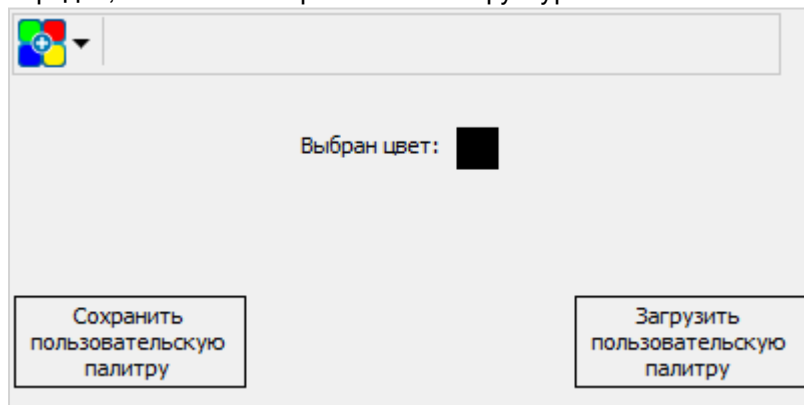
К проекту подключен внешний модуль **CommonLib**. Проект содержит единственную форму *MainForm*. На ней расположены панель инструментов *ToolBar* и прямоугольник *Output*, окрашивающийся в цвет, выбранный с помощью панели инструментов. К панели инструментов *ToolBar* применен инструмент линейной разметки. Также на форме расположен вспомогательный экземпляр *Files* компонента **Окружение: файлы**, служащий для сохранения и открытия файлов.

На панели инструментов размещены:

- кнопка *OpenColorDialog* - кнопка типа [2.1.11.1. Кнопка панели инструментов \(стр. 44\)](#), по щелчку ЛКМ на которую открывается диалоговое окно для выбора цвета из палитры [2.2.5. Диалог выбора цвета \(стр. 78\)](#);
- кнопка *OpenColorMenu* - кнопка типа [2.1.11.2. Кнопка выпадающего списка \(стр. 46\)](#), по щелчку ЛКМ на которую открывается выпадающее меню для выбора цвета из списка;
- кнопки *ButtonSave* и *ButtonOpen*, предназначенные для сохранения и загрузки пользовательской палитры цветов, выбранной в окне палитры [2.2.5. Диалог выбора цвета \(стр. 78\)](#);
- разделитель элементов типа [2.1.11.3. Разделитель панели инструментов \(стр. 47\)](#).



Запустите проект-пример в рантайм. Обратите внимание, что элементы панели расположены в таком порядке, в каком они перечислены в структуре объекта.



Щелкните ЛКМ по кнопке выбора цвета из палитры (*OpenColorDialog*, ). Активируется событие

MouseClick кнопки, открывающее [2.2.5. Диалог выбора цвета \(стр. 78\)](#) в новом окне. Ознакомьтесь с обработчиком события.

- Свойство **Result** компонента [2.2.5. Диалог выбора цвета \(стр. 78\)](#) записывает результат выбора цвета в экземпляре [2.1.13.2. Результат выбора цвета \(стр. 55\)](#), расположенный в структуре объекта внутри *OpenColorDialog*.
- Свойство **okAction** компонента [2.2.5. Диалог выбора цвета \(стр. 78\)](#) ссылается на команду *onClick*, расположенную в структуре объекта внутри *OpenColorDialog*. Команда выполняется в момент нажатия кнопки **OK** в окне [2.2.5. Диалог выбора цвета \(стр. 78\)](#) и окрашивает прямоугольник *Output* в цвет, записанный в *ColorDialogResult*.

Выберите цвет из палитры и нажмите ОК. Прямоугольник окрасится в выбранный цвет.

Теперь щелкните ЛКМ по кнопке, открывающей выпадающее меню (*OpenCoLoMenu*, ). Активируется

событие **MouseClicked** кнопки, открывающее [2.2.5. Диалог выбора цвета \(стр. 78\)](#) в новом окне. Ознакомьтесь с обработчиком события.

- Свойство **Items** компонента [2.2.1. Меню \(стр. 70\)](#) ссылается на функцию **GetMenuItems()**, расположенную в структуре объекта внутри *OpenCoLoMenu*. Функция описывает пункты меню в виде JSON массива. Ознакомьтесь с описанием функции.
- Свойство **pResult** компонента [2.2.1. Меню \(стр. 70\)](#) записывает id выбранного пункта меню в экземпляр [2.1.10.3. Результат выбора меню \(стр. 43\)](#), расположенный в структуре объекта внутри *OpenCoLoMenu*.
- Свойство **pAction** компонента [2.2.1. Меню \(стр. 70\)](#) ссылается на команду *onClick*, расположенную в структуре объекта внутри *OpenCoLoMenu*. Команда выполняется в момент выбора пункта меню и окрашивает прямоугольник *Output* в цвет, соответствующий id выбранного пункта меню.

Выберите цвет из меню. Прямоугольник окрасится в выбранный цвет.

Сохраним пользовательскую палитру в файл.

1. Откройте окно выбора цвета вновь и добавьте несколько цветов к пользовательской палитре.
2. Закройте окно выбора цвета и нажмите **Сохранить пользовательскую палитру**. Активируется событие **ButtonPressed** кнопки. Ознакомьтесь с обработчиком события. Экземпляр *Files* компонента **Окружение**: файлы формирует путь к файлу `colors.json`, создаваемом в папке проекта `output`. Затем туда записывается пользовательская палитра в виде JSON-строки из свойства **CustomColors** компонента [2.2.5. Диалог выбора цвета \(стр. 78\)](#).
3. Убедитесь, что файл создан в указанной папке.

Теперь откроем сохраненную пользовательскую палитру из файла.

1. Закройте окно проекта-примера, а затем вновь запустите в рантайм. Откройте окно выбора цвета: пользовательская палитра будет пуста.
2. Закройте окно выбора цвета и нажмите **Загрузить пользовательскую палитру**. Активируется событие **ButtonPressed** кнопки. Ознакомьтесь с обработчиком события. Экземпляр *Files* компонента **Окружение**: файлы извлечет содержимое файла `colors.json` и поместит его в компонент [2.2.5. Диалог выбора цвета \(стр. 78\)](#) в качестве значения свойства **CustomColors**.
3. Откройте окно выбора цвета и убедитесь, что пользовательская палитра обновилась.

2. Справочное руководство

2.1. Типы

2.1.1. Тема оформления

Компонент, позволяющий менять тему оформления визуальных компонентов SePlatform.HMI.CommonLib: шрифты, цвет текста, фона и т.п.. Подробнее - в свойствах компонента.

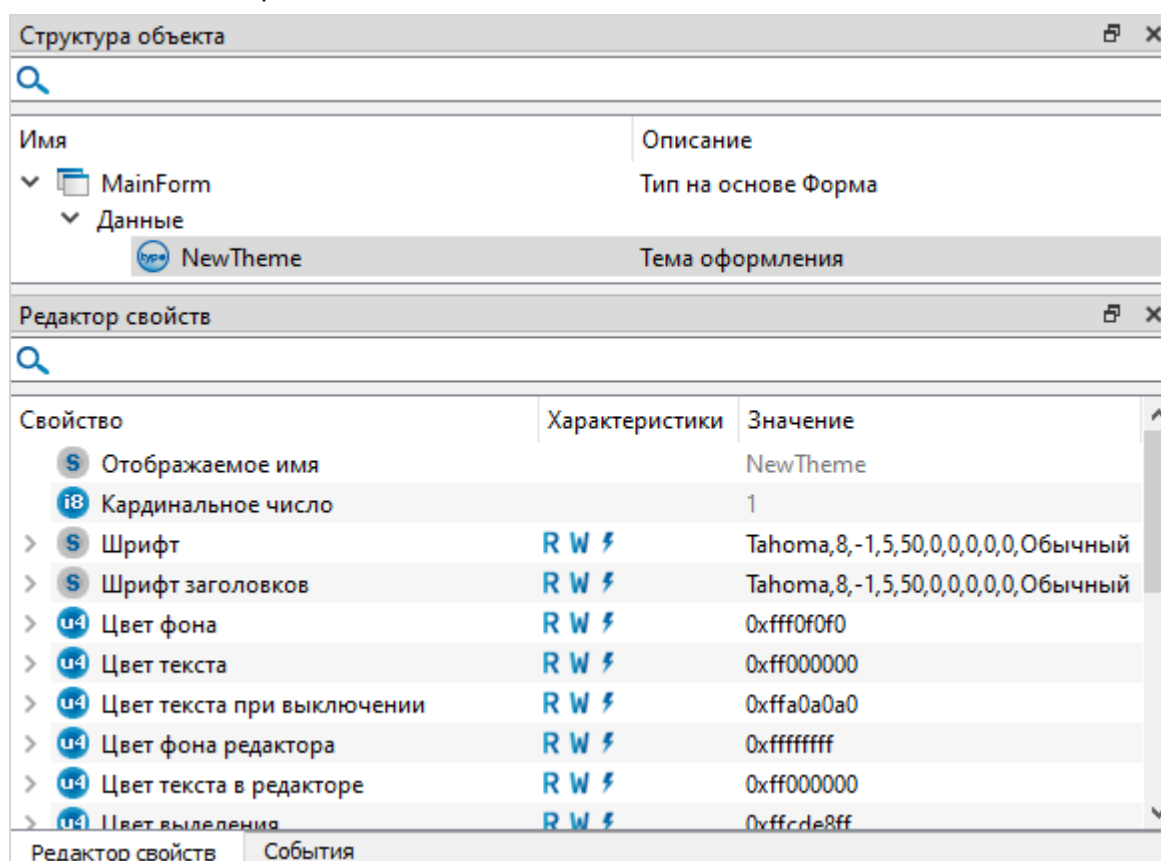
Для оформления визуальных компонентов, добавленных на панель инструментов, используйте компонент [2.1.11.5. Тема оформления панели инструментов \(стр. 49\)](#).

Экземпляр этого типа не отображается на форме и виден только в области **Структура объекта**.

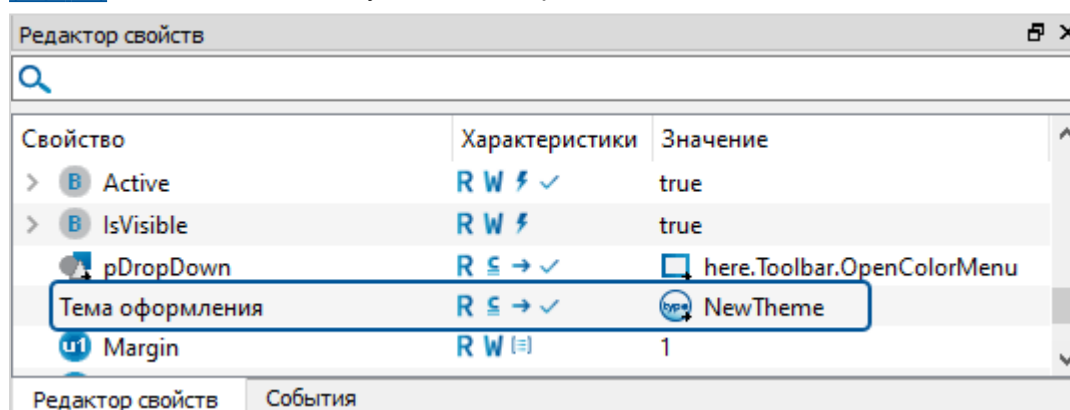


ПРИМЕР

NewTheme - экземпляр типа *Theme*.



Чтобы применить тему *NewTheme* к объекту, укажите в свойстве [2.3.1. Тема оформления \(pTheme\)](#) (стр. 97) этого объекта ссылку на экземпляр *NewTheme*.



ПРИМЕЧАНИЕ

На экземпляре может быть сделано сколько угодно ссылок.

2.1.1.1. Свойства

Шрифт (Font)

Шрифт основного текста. Тип значения - string.

Шрифт заголовков (TitleFont)

Шрифт заголовков. Тип значения - string.

Цвет фона (BackgroundColor)

Цвет основной области компонента. Тип значения - уведомляющий uint4.

Цвет текста (ForegroundColor)

Цвет текста в основной области компонента. Тип значения - уведомляющий uint4.

Цвет текста при выключении (DisabledFontColor)

Цвет текста неактивных элементов. Тип значения - уведомляющий uint4.

Цвет фона редактора (EditorBrushColor)

Цвет фона области компонента, в которой можно изменить значение. Тип значения - уведомляющий uint4.

Цвет текста в редакторе (EditorFontColor)

Цвет текста в области компонента, в которой можно изменить значение. Тип значения - уведомляющий uint4.

Цвет выделения (SelectionColor)

Цвет активного (выбранного) элемента в области компонента, в которой можно изменить значение. Тип значения - уведомляющий uint4.

Цвет рамки (BorderColor)

Цвет рамки, ограничивающей область компонента, в которой можно изменить значение. Тип значения - уведомляющий uint4.

Цвет рамки при выключении (DisabledBorderColor)

Цвет рамки, ограничивающей область компонента, в которой можно изменить значение, если эта область неактивна. Тип значения - уведомляющий uint4.

Цвет фона панели инструментов (ToolbarBrushColor)

Цвет фона панели инструментов. Тип значения - уведомляющий uint4.

Цвет текста на панели инструментов (ToolbarFontColor)

Цвет текста на панели инструментов. Тип значения - уведомляющий uint4.

Цвет редактора на панели инструментов (ToolbarEditorColor)

Цвет фона расположенного на панели инструментов компонента, в котором можно изменить значение. Тип значения - уведомляющий uint4.

Цвет выделения на панели инструментов (ToolbarSelectionColor)

Цвет выделения элемента на панели инструментов. Тип значения - уведомляющий uint4.

Цвет фона меню (MenuBrushColor)

Цвет фона меню. Тип значения - уведомляющий uint4.

Цвет текста в меню (MenuFontColor)

Цвет текста в меню. Тип значения - уведомляющий uint4.

Цвет текста недоступного элемента меню (MenuDisabledFontColor)

Цвет текста недоступного элемента меню. Тип значения - уведомляющий uint4.

Цвет выделенного элемента меню (MenuSelectionColor)

Цвет выделенного элемента меню. Тип значения - уведомляющий uint4.

2.1.1.2. Функции

GetViewSettings

```
string GetViewSettings()
```

Возвращает настройки текущей темы в виде json-строки. Эти настройки могут использоваться для восстановления внешнего вида компонентов с помощью функции [UpdateView\(\)](#) при перезапуске проекта.



ПРИМЕР

Пример возвращаемого значения:

```
{
  "Background": "0xffff0f0f",
  "Border": "0xff808080",
  "DisabledBorder": "0xffa0a0a0",
  "DisabledFont": "0xff808080",
  "EditorBrush": "0xffffffff",
  "EditorFont": "0xff000000",
  "Font": "Tahoma,8,-1,5,50,0,0,0,0,Обычный",
  "Foreground": "0xff000000",
  "MenuBrushColor": "0xffff0f0f",
  "MenuDisabledFontColor": "0xff808080",
  "MenuFontColor": "0xff000000",
  "MenuSelectionColor": "0xffcde8ff",
  "Selection": "0xffcde8ff",
  "TitleFont": "Tahoma,8,-1,5,50,0,0,0,0,Обычный",
  "ToolbarBrushColor": "0xffff0f0f",
  "ToolbarEditorColor": "0xffffffff",
  "ToolbarFontColor": "0xff000000",
  "ToolbarSelectionColor": "0xffcde8ff"
}
```

UpdateView

```
void UpdateView(string json)
```

Меняет внешний вид компонентов, у которых в свойстве [2.3.1. Тема оформления \(pTheme\) \(стр. 97\)](#) указана текущая тема, в соответствии с настройками этой темы. Настройки могут быть получены в результате вызова функции [GetViewSettings\(\)](#).

Входной параметр - настройки текущей темы в виде json-строки.

2.1.2. Конвертер значений

Компонент, использующийся для различных преобразований значений.

Экземпляр этого типа не отображается на форме и виден только в области **Структура объекта**.

Чтобы выполнить преобразование, вызовите функцию компонента: `Converter.Function()`.

2.1.2.1. Функции

Convert

```
variant Convert(variant value, string dataType)
```

Конвертирует значение `value` типа `variant` в значение типа `dataType`. Результат функции хранится в переменной типа `variant`.

В качестве значения `dataType` можно указывать только названия типов данных языка SePlatform.Om. Типы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции CanonicalTypeToDataType ([стр. 20](#)) (см. столбец **Название типа**). Также можно использовать значение, возвращаемое этой функцией.

Входные параметры:

- конвертируемое значение;
- тип данных, в который конвертируется значение.

Преобразование выполняется с учетом ограничений пространства имен TypeConvert. Подробнее об ограничениях - в документе на SePlatform.Om.



ПРИМЕР

```
number: variant = Converter.Convert(52, "uint4")
```

StringToVariant

```
variant StringToVariant(string value, string dataType)
```

Конвертирует строковое значение `value` в значение типа `dataType`. Результат функции хранится в переменной типа `variant`.

В качестве значения `dataType` можно указывать только названия типов данных языка SePlatform.Om. Типы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции CanonicalTypeToDataType ([стр. 20](#)) (см. столбец **Название типа**). Также можно использовать значение, возвращаемое этой функцией.

Входные параметры:

- конвертируемое значение;
- тип данных, в который конвертируется значение.



ПРИМЕР

```
number: variant = Converter.StringToVariant("52", "uint4")
```

VariantToString

```
string VariantToString(variant value, string dataType)
```

Конвертирует переменную типа `variant`, хранящую значение `value` типа `dataType`, в значение типа `string`.

В качестве значения `dataType` можно указывать только названия типов данных языка `SePlatform.Om`. Типы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции `CanonicalTypeToDataType` ([стр. 20](#)) (см. столбец **Название типа**). Также можно использовать значение, возвращаемое этой функцией.

Входные параметры:

- конвертируемое значение;
- тип данных конвертируемого значения.



ПРИМЕР

```
Number.Text = Converter.VariantToString(52, "uint4")
```

VariantToLocalizedString

```
string VariantToLocalizedString(variant value, string dataType)
```

Конвертирует переменную типа `variant`, хранящую значение `value` типа `dataType`, в значение типа `string` с учетом региональных настроек. В региональных настройках могут быть заданы точность (количество знаков после разделителя) и знак разделителя.

В качестве значения `dataType` можно указывать только названия типов данных языка `SePlatform.Om`. Типы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции `CanonicalTypeToDataType` ([стр. 20](#)) (см. столбец **Название типа**). Также можно использовать значение, возвращаемое этой функцией.

Входные параметры:

- конвертируемое значение;
- тип данных конвертируемого значения.



ПРИМЕР

```
Text_1 = String.VariantToLocalizedString(a, "double");
```

ValidateValue

```
string ValidateValue(string value, string dataType)
```

Проверяет значение `value`, записанное в виде строки, на соответствие указанному типу данных `dataType`.

Входные параметры:

- проверяемое значение;
- проверяемый тип данных.

В качестве значения `dataType` можно указывать только названия типов данных языка `SePlatform.Om`. Типы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции `CanonicalTypeToDataType` ([стр. 20](#)) (см. столбец **Название типа**). Также можно использовать значение, возвращаемое этой функцией.

Возвращает:

- текст ошибки в случае ее возникновения;
- пустую строку, если ошибки не возникает.



ПРИМЕР

Вызов: `Converter.ValidateValue("1234", "uint4")`

ValidateRange

```
string ValidateRange(variant value, string dataType, variant min, variant max)
```

Проверяет, входит ли указанное значение `value` типа `variant` в допустимый диапазон от `min` до `max`.

Входные параметры:

- проверяемое значение;
- тип данных проверяемого значения;
- нижняя граница диапазона `min`;
- верхняя граница диапазона `max`.

Если не указать значения `min` и/или `max`, будут использоваться границы диапазона указанного типа данных.

В качестве значения `dataType` можно указывать только названия типов данных языка `SePlatform.Om`. Типы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции `CanonicalTypeToDataType` ([стр. 20](#)) (см. столбец **Название типа**). Также можно использовать значение, возвращаемое этой функцией.

Возвращает:

- текст ошибки в случае ее возникновения;
- пустую строку, если ошибки не возникает.



ПРИМЕР

Вызов: `Converter.ValidateRange(123, "uint8", 1, 500)`

ValidateValueRange

```
string ValidateValueRange(string value, string dataType, variant min, variant max)
```

Проверяет значение `value`, указанное в виде строки, на соответствие указанному типу данных `dataType` и на принадлежность допустимому диапазону от `min` до `max`.

Входные параметры:

- проверяемое значение;
- проверяемый тип данных;
- нижняя граница диапазона `min`;
- верхняя граница диапазона `max`.

В качестве значения `dataType` можно указывать только названия типов данных языка `SePlatform.Om`. Типы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции `CanonicalTypeToDataType` ([стр. 20](#)) (см. столбец **Название типа**). Также можно использовать значение, возвращаемое этой функцией.

Возвращает:

- текст ошибки в случае ее возникновения;
- пустую строку, если ошибки не возникает.



ПРИМЕР

Вызов: `Converter.ValidateValueRange(123, "uint8", 1, 500)`

CanonicalTypeToDataType

```
string CanonicalTypeToDataType(uint1 canonicalType)
```

Вспомогательная функция. Возвращает название канонического типа данных, указанного в числовом виде. Возвращаемое значение можно использовать в качестве входного параметра `dataType` остальных функции `Convert`.

Входной параметр - число, соответствующее каноническому типу данных.

**Таблица соответствия
численного и текстового
названий канонического
типа данных**

Число типа	Название типа
0	"void"
1	"bool"
2	"int1"
3	"uint1"
4	"int2"
5	"uint2"
6	"int4"
7	"uint4"
8	"int8"
9	"uint8"
10	"float"
11	"double"

Число типа	Название типа
12	"timestamp"
14	"string"

**ПРИМЕР**

Вызов: `a = Converter.CanonicalTypeToDataType(3)`

Результат: переменная `a` типа `string`, значение - «`uint1`».

2.1.3. Утилиты для работы со строками

Компонент, использующийся для различных преобразований строковых значений.

Экземпляр этого типа не отображается на форме и виден только в области **Структура** объекта.

Чтобы выполнить преобразование, вызовите функцию компонента: `StringUtils.Function()`.

2.1.3.1. Функции

GetDeclension

```
string GetDeclension(int8 number, string nominative, string genitive, string plural)
```

Склоняет указанное во входных параметрах слово в зависимости от указанного числа (количества).

Входные параметры:

- число (количество), в зависимости от которого склоняется слово;
- склоняемое слово в именительном падеже;
- склоняемое слово в родительном падеже;
- склоняемое слово во множественное число.

Возвращает слово в склонении, подходящем указанному числу.

**ПРИМЕР**

Предположим, некая переменная `hour` хранит количество часов, оставшихся до конца сессии пользователя, а в переменную `word` нужно записывать склоняемое слово "час". Тогда:

```
word: string = StringUtils.GetDeclension(hour, "час", "часа", "часов");  
Message.Text = "Осталось " + hour + word;
```

CalcRowHeight

```
int4 CalcRowHeight(string font)
```

Вычисляет примерную высоту строки текста для указанного шрифта `font`. Применяется для определения высоты заголовка в таблице при изменении шрифта темы [2.1.1. Тема оформления \(стр. 12\)](#).

Входным значением является строка, в которой перечислены параметры шрифта.

**ПРИМЕР**

Пример строки входного значения:

```
Tahoma, 8, -1, 5, 50, 0, 0, 0, 0, Обычный
```

2.1.4. Утилиты для работы с окном

Вспомогательный тип для работы с окнами (формами в режиме исполнения).

Экземпляр этого типа не отображается на форме и виден только в области **Структура объекта**.

2.1.4.1. Свойства

Ссылка на форму (pForm)

Ссылка на форму, поведение окна которой настраивается с помощью функций этого компонента.

2.1.4.2. Функции

FitFormSize

```
void FitFormSize(double windowWidth, double windowHeight)
```

Подстраивает размер формы под размер окна, в котором она открывается, с учетом масштаба формы.

Входные параметры - ширина и высота окна.

CorrectPosition

```
void CorrectPosition()
```

Корректирует положение окна так, чтобы оно отобразилось на первом мониторе, если:

- окно не отображается ни на одном из имеющихся мониторов;
- окно невозможно переместить из-за выхода заголовка окна за пределы монитора.

Применяется при открытии окна с определенными координатами, которые могут выходить за область текущего монитора.

FitToMonitor

```
void FitToMonitor(int4 monitor)
```

Размещает окно на указанном мониторе. Входной параметр - номер монитора, который можно получить с помощью функции [GetMonitor\(\)](#).

CorrectPositionAboutPoint

```
void CorrectPositionAboutPoint(int4 x, int4 y, int4 pointHeight)
```

Корректирует положение окна относительно указанной точки так, чтобы окно располагалось в пределах монитора. Применяется при открытии контекстного меню и других всплывающих окон.

Входные параметры:

- X-координата точки;
- Y-координата точки;
- смещение вверх относительно низа монитора.

GetMonitor

```
int4 GetMonitor()
```

Возвращает номер монитора, на котором отображается большая часть окна. Если окно не отображается ни на одном из имеющихся мониторов, то функция возвращает значение «-1».

Можно использовать совместно с функцией [FitToMonitor\(\)](#), чтобы расположить окно целиком на том мониторе, где открывается его большая часть.

GetMonitorForRect

```
int4 GetMonitorForRect(double left, double top, double width, double height)
```

Возвращает номер монитора, на котором отображается большая часть родительского компонента. Может использоваться для открытия формы в центре определенного прямоугольника.

Входные параметры:

- X-координата левого верхнего угла родительского компонента;
- Y-координата левого верхнего угла родительского компонента;
- ширина родительского компонента;
- высота родительского компонента.

2.1.5. Информация о продукте

Вспомогательный тип. Содержит информацию о продукте, компании-разработчике и платформе. Позволяет получить эту информацию при подключении продукта к проекту SePlatform.HMI в качестве внешнего модуля.

При разработке собственного продукта экземпляр типа следует помещать в глобальные объекты проекта.

2.1.5.1. Свойства

CompanyName

Название компании-разработчика продукта. Тип значения - string.

CompanyTag

Тег компании. Тип значения - string.

PlatformName

Название платформы. Тип значения - string.

PlatformTag

Тег платформы. Тип значения - string.

FamilyName

Название семейства продуктов. Тип значения - string.

FamilyTag

Тег семейства продуктов. Тип значения - string.

ProductName

Название продукта. Тип значения - string.

ProductTag

Тег продукта. Тип значения - string.

FolderName

Имя папки для хранения данных продукта. Обычно совпадает с тегом компании. Тип значения - string.

Version

Версия продукта. Тип значения - string.

2.1.5.2. Функции

GetProgramDataFolder

```
string GetProgramDataFolder()
```

Предоставляет полный путь к папке, предназначенной для хранения данных программы - например, конфигурационных файлов, бэкапов и пр. Путь формируется из значений свойств:

```
Files.CombinePath(Files.AllUsersFolder, Files.CombinePath(FolderName, ProductName))
```

где FolderName и ProductName - свойства типа **ProductInfo**.

GetUserDataFolder

```
string GetUserDataFolder()
```

Предоставляет полный путь к папке, предназначенной для хранения пользовательских данных - например, настроек вида программы для конкретного пользователя. Путь формируется из значений свойств:

```
Files.CombinePath(Files.CurrentUserFolder, Files.CombinePath(FolderName, ProductName))
```

где FolderName и ProductName - свойства типа **ProductInfo**.

2.1.6. Буфер аргументов события

Вспомогательный тип. Представляет собой буфер, предназначенный для передачи аргументов события, реализованного с помощью компонента **Команда (Action)**.

Запись и извлечение данных происходит по принципу FIFO¹ (очереди). При переполнении буфера новые аргументы вытесняют старые также по принципу очереди.

Пример использования

Для демонстрации использования компонента разработан проект-пример. Проект-пример можно скачать и открыть в Дизайнер SePlatform.HMI. Для работы с проектом-примером потребуются компоненты, указанные в [1.2. Подготовка к работе \(стр. 6\)](#), и SePlatform.Data Server с подключенным модулем логики.

¹first-in-first-out - "первым пришел, первым ушел"

Предположим, необходимо получить данные о сигналах из SePlatform.Data Server в проекте SePlatform.HMI посредством JSON-запросов к API сервера. В результате получения необходимых данных должна выполняться определенная команда. Запрос по каждому сигналу происходит отдельно. Таких запросов может быть достаточно много. Поскольку запросы происходят асинхронно, к началу обработки результатов может быть получено несколько ответов. Чтобы не потерять результаты запросов, каждый из них помещается в буфер типа **EventArgs**, который по очереди передает полученные данные в команду в качестве аргумента.

Откройте форму **MainForm** в проекте-примере. На форме размещена кнопка *Получить ID* (**ButtonGetIDs**). Ознакомьтесь с кодом обработчика события **ButtonPressed** кнопки.

```
ServerAPI.GetID("Service.State.Pair");
ServerAPI.GetID("Service.State.Server");
ServerAPI.GetID("Service.Config.Version");
//по нажатию кнопки вызывается функция GetID типа ServerAPI
```

По нажатию кнопки вызывается функция **GetID()** компонента *ServerAPI*, разработанного для данного проекта-примера. В качестве входных параметров указаны имена сигналов SePlatform.Data Server. Поскольку такие - сервисные - сигналы присутствуют в любой конфигурации, они выбраны для этого примера. Здесь можно указать любые имена сигналов, присутствующих в вашем сервере.

Обратите внимание на команду *OnGetID* здесь же, в компоненте *ServerAPI*. Это команда, выполняемая при получении ID сигнала из сервера. Она записывает полученное значение в текстовое поле *TagsIDs*. Именно в эту команду необходимо передавать аргумент - полученный ID сигнала.

Откройте тип **ServerAPI**. В типе созданы:

- функция **GetID()**, отправляющая JSON-запрос на получение ID по имени сигнала на сервер;
- элемент *InvokeFromJSON*, связанный с динамическим сигналом сервера **InvokeFromJSON**, куда функция **GetID()** отправляет JSON-запрос, а сервер помещает JSON-ответ;
- *pOnGetID* - ссылка на команду *OnGetID*, реализованную в **MainForm**, выполняемую в результате получения ID сигнала;
- буфер *OnGetIDArgs*, предназначенный для передачи полученного ID из ссылки на команду *pOnGetID* в команду *OnGetID*. Размер буфера ограничен десятью записями. Размер буфера можно изменить в свойстве **BufferSize**.

Ознакомьтесь с функцией **GetID()**. Функция использует API предоставления данных сервера. Подробнее API и работа с ним описаны в документе на модуль логики SePlatform.Data Server.

```
transac += 1;
//наращиваем номер транзакции (обращения к серверу)
request: string = "{\"transaction\":\""+String.ToString(
(transac)+"\", \"request\":
{ \"target\":\"Service.ServerApi\", \"method\":\"GetIdByTagName\", \"input\":
{ \"tagname\":\""+tagname+"\" } } }";
//в переменной request формируется запрос в формате JSON
InvokeFromJSON.TrySetValue(request);
//запрос помещается в компонент, связанный с динамическим сигналом сервера
InvokeFromJSON
//сервер помещает ответ в формате JSON в этот же сигнал
```

В случае успешной записи ответа сервера в компонент *InvokeFromJSON* активируется событие **OnValueChanged** этого компонента. Ознакомьтесь с обработчиком этого события.

```

var ask = JSON.parse(value);
//при изменении значения сигнала - то есть, при получении ответа от сервера -
//полученное значение записывается в переменную ask, где его необходимо
разобрать
var res = ask.result.return;
//в переменную res записывается значение ключа "return" - это и есть ID сигнала
OnGetIDArgs.Push(res);
//в буфер типа EventArgs записывается полученное значение - ID (иначе -
аргумент)
pOnGetID.Invoke();
//вызывается команда, в которую передается полученный аргумент

```

Полученный от сервера JSON-ответ необходимо разобрать с помощью функции **JSON.parse()**, и затем выделенный из ответа ID поместить в буфер *OnGetIDArgs* с помощью функции **Push()**. После этого с помощью функции **Invoke()** по ссылке *pOnGetID* вызывается команда *OnGetID*, куда полученный ID передается через буфер.



ОБРАТИТЕ ВНИМАНИЕ

Здесь разбирается внутренний аргумент события **OnValueChanged** компонента *InvokeFromJSON* - value, а не значение свойства **Value** этого компонента. Это тоже связано с асинхронностью процессов получения ответов от сервера.

Значение свойства **Value** перезаписывается при получении каждого нового значения, а в аргументе value полученное значение сохраняется до передачи в обработчик события. Так как процесс обработки асинхронный, то к моменту выполнения обработчика значение свойства **Value** может несколько раз измениться. А значение аргумента value будет тем же, что и в момент наступления события.

Вернитесь на форму **MainForm** и ознакомьтесь с обработчиком события **Invoked** команды *OnGetID*.

```

var getResult = ServerAPI.OnGetIDArgs.Get();
//переменная getResult создается для извлечения аргумента события из буфера
TagsIDs.Text = TagsIDs.Text + " " + getResult;
//значение переменной записывается в текстовое поле

```

Здесь с помощью функции **Get()** происходит извлечение аргумента события из буфера *OnGetIDArgs*. Затем это значение используется для записи в текстовое поле на форме. Таким образом из буфера извлекается каждый полученный ID в момент получения ответа от сервера.

2.1.6.1. Свойства

Размер буфера (BufferSize)

Размер буфера аргументов, измеряемый количеством записей. Тип значения - int4.

2.1.6.2. Функции

Push

```
void Push(string args)
```

Записывает аргумент в буфер. Запись помещается в конец очереди.

Get

```
string Get()
```

Извлекает первый в очереди аргумент из буфера.

2.1.7. Обработчик события

Компонент позволяет создать пользовательское событие с возможностью передачи его аргументов. Реализован на базе компонента **Команда (Action)** и буфера аргументов **Args** типа [2.1.6. Буфер аргументов события \(стр. 25\)](#).

Компонент можно использовать, например, когда необходимо, чтобы аргументы события, произошедшего на одной форме проекта, были переданы на другую форму в момент наступления события. Для этого:

1. На форме, ожидающей передачи аргументов события, создается экземпляр обработчика события *EventHandler*, а на форме, где ожидается наступление события и генерация его аргументов - ссылка на обработчик события *pEventHandler*.
2. Когда нужно уведомить *EventHandler* о наступлении события:
 - 2.1. В экземпляр *pEventHandler* помещается аргумент события. Для этого используется вложенный объект *pEventHandler.Args*. Аргумент помещается в буфер с помощью команды:

```
pEventHandler.Args.Push(string arg)
```

- 2.2. Затем вызывается функция **Invoke()** экземпляра *pEventHandler*, активирующая событие *EventHandler*.

3. В обработчике события **Invoked** экземпляра *EventHandler*:

- 3.1. Сначала извлекается переданный аргумент с помощью команды:

```
EventHandler.Args.Get()
```

- 3.2. Затем выполняется обработка полученного аргумента.

Для корректной работы буфера с аргументами следует соблюдать следующие правила:

- один раз вызывать функцию **Push()** перед вызовом функции **Invoke()** экземпляра *pEventHandler* при формировании события;
- один раз вызывать функцию **Get()** при получении аргументов события в обработчике *EventHandler*.

2.1.7.1. Свойства

Args

Объект, представляющий собой буфер аргументов события. Тип объекта - [2.1.6. Буфер аргументов события \(стр. 25\)](#).

Запись и извлечение данных в буфер происходит по принципу FIFO¹ (очереди). При переполнении буфера новые аргументы вытесняют старые также по принципу очереди.

Имеет собственные свойства и функции, аналогичные свойствам и функциям компонента [2.1.6. Буфер аргументов события \(стр. 25\)](#).

Включено (Enabled)

Включает или отключает команду. Тип значения - bool.

Значения:

- «true» - команда включена и может выполняться;
- «false» - команда отключена и не может выполняться.

Триггер (InvokeTrigger)

Свойство-триггер команды. Тип значения - bool. Значение по умолчанию - «false», при записи значения «true» команда выполняется.

Размер буфера аргументов (ArgsBufferSize)

Размер буфера аргументов, измеряемый количеством записей. Тип значения - int4.

2.1.7.2. Функции

Invoke

```
void Invoke()
```

Запускает пользовательскую команду, указанную в обработчике события **Invoked**.

2.1.7.3. События

Invoked

Активируется в результате вызова функции **Invoke()** или записи значения «true» в свойство **InvokeTrigger**.

¹first-in-first-out - "первым пришел, первым ушел"

EnabledChanged

Активируется в момент смены состояния команды, указанного в значении свойства **Включено (Enabled)**.

2.1.8. Текстовые шаблоны

Вспомогательный тип, представляющий собой набор текстовых шаблонов, которые могут быть подставлены, например, в текст колонтитула в [2.2.7. Редактор колонтитула \(стр. 82\)](#).

Принцип использования шаблонов в текстах колонтитулов приведен в описании компонента [2.2.7. Редактор колонтитула \(стр. 82\)](#).



ОБРАТИТЕ ВНИМАНИЕ

Для использования шаблона в тексте нужно указать имя шаблона, обрамленное в специальные символы `${TemplateName}`, где `TemplateName` - это имя шаблона. Пример приведен в описании функции `Expand()` компонента [\(стр. 31\)](#).

2.1.8.1. Свойства

pAdditionData

Ссылка на таблицу, содержащую дополнительные текстовые шаблоны. Используется для объединения двух наборов шаблонов.

Дополнительные шаблоны могут быть добавлены из другого экземпляра компонента **Текстовые шаблоны**. Для этого в качестве значения свойства основного экземпляра укажите «`Template.Data`», где «`Template`» - имя другого экземпляра компонента **Текстовые шаблоны**, «`Data`» - обращение к данным таблицы (шаблонам) этого экземпляра.

pValueChanged

Ссылка на команду, выполняемую при изменении значения любого из шаблонов.

2.1.8.2. Функции

Set

```
void Set(string name, variant value)
```

Устанавливает для шаблона `name` значение `value`. Имя шаблона указывается произвольно. Если значение указанного шаблона уже было задано ранее, оно будет обновлено.



ПРИМЕР

Примеры создания шаблонов:

```
Templates.Set("Department", "Участок 1");
```

```
Templates.Set("CurrentUser", CurrentUserDisplayName);
```

```
Templates.Set("Now", DateTime.ToString(DateTime.Now()));
```

Clear

```
void Clear()
```

Очищает набор шаблонов.

Expand

```
string Expand(string text)
```

Возвращает текст, в котором названия шаблонов заменены их значениями. Шаблон следует вызывать, используя специальные символы: `${TemplateName}`, где `TemplateName` - это имя шаблона. Если указать не существующий или не имеющий значения шаблон, то результирующий текст будет содержать обозначение шаблона без изменений.



ПРИМЕР

Пусть с помощью функции **Set()** в набор шаблонов *Templates* помещен шаблон *Department*, содержащий название участка производства.

```
Templates.Set("Department", "Участок 1");
```

Предположим, в заголовке *Header* необходимо указать название участка из шаблона. Тогда вызов шаблона во входном параметре функции **Expand()** выглядит следующий образом:

```
Header.Text=Templates.Expand("Указаны объекты для ${Department}");
```

В результате в заголовок *Header* будет записан текст "Указаны объекты для Участок 1".

ToJSON

```
string ToJSON()
```

Предоставляет перечень шаблонов, добавленных в текущий набор, в формате JSON.

Может использоваться для сохранения текущего набора в файл, создаваемый для последующей загрузки в экземпляр компонента.

Функция не учитывает дополнительные текстовые шаблоны, которые могут быть указаны в свойстве `pAdditionData` ([стр. 30](#)). Чтобы получить полный набор шаблонов, используйте функцию `GetTemplateList` ([стр. 32](#)).

**ПРИМЕР**

Пример значения, возвращаемого функцией:

```
[{"name": "Now", "value": "30.11.2023 15:20:39"},  
{"name": "CurrentUser", "value": "Иванов"}, {"name": "Department", "value": "Участок 1"}]
```

FromJSON

```
void FromJSON(string json)
```

Разбирает указанный в формате JSON набор шаблонов. Используется для загрузки шаблонов в экземпляр компонента.

GetTemplateList

```
string GetTemplateList()
```

Предоставляет перечень шаблонов, добавленных в текущий набор, в формате JSON, включая дополнительные наборы, указанные в свойстве `pAdditionData` ([стр. 30](#)). Чтобы получить набор шаблонов компонента без дополнительных наборов, используйте функцию `ToJSON` ([стр. 31](#)).

**ПРИМЕР**

Пример значения, возвращаемого функцией:

```
[{"name": "ExtraTemplate", "value": "Дополнительная информация"},  
{"name": "Now", "value": "30.11.2023 15:20:39"}, {"name": "CurrentUser", "value": "Иванов"},  
{"name": "Department", "value": "Участок 1"}]
```

2.1.9. Редакторы

2.1.9.1. Редактор целого числа

Компонент, позволяющий менять значения целочисленных переменных с указанным шагом.

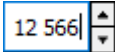
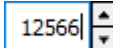


Свойства

Localized

Позволяет указать, должны ли числа в редакторе отображаться с учетом знака разделителя групп разрядов, указанного в региональных настройках. Тип значения - bool.

Значение:

- «true» - число отображается с учетом знака разделителя: 
- «false» - число отображается без учета знака разделителя: 

Значение (Value)

Изменяемое значение. Тип значения - уведомляющий int8.

Максимум (Maximum)

Максимум, до которого можно инкрементировать изменяемое значение. Тип значения - int8.

Минимум (Minimum)

Минимум, до которого можно декрементировать изменяемое значение. Тип значения - int8.

Step (Шаг изменения)

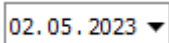
Шаг инкремента / декремента. Тип значения - int8.

ValueChanged

Ссылка на команду, выполняемую при изменении значения.

2.1.9.2. Редактор даты

Окно редактора даты для ввода даты в виде «день : месяц : год».



Значения полей редактора можно менять:

- с помощью ввода цифр;
- клавишами-стрелками;
- выбрав дату из выпадающего календаря ([стр. 73](#)).



ПРИМЕЧАНИЕ

Можно использовать совместно с редактором времени ([стр. 34](#)).

Свойства

Значение (Value)

Хранит введенные в окне редактора дату и время, округленное до секунд. Тип значения - уведомляющий timestamp.

При редактировании меняет значение даты, а значение времени оставляет без изменений.

pValueChanged

Ссылка на команду, выполняемую при изменении даты в окне редактора.



ПРИМЕР

При вводе даты в окне редактора проверяется условие: если введенная дата больше текущей, появляется сообщение "Этот день еще не настал", в противном случае - "Дата выбрана". Тогда код обработчика команды:

```
DateEditor.Error = DateEditor.Value > DateTime.Now() ? "Этот день еще не настал" :  
"Дата выбрана";
```

Описание ошибки (Error)

Текст ошибки ввода даты. Тип значения - уведомляющий string.

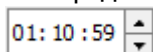


ПРИМЕЧАНИЕ

Свойство не предназначено для проверки введенной даты. Проверку даты можно описать в команде, на которую ссылается свойство **pValueChanged**.

2.1.9.3. Редактор времени

Окно редактора времени для ввода времени в виде «часы:минуты:секунды».



Значения полей редактора можно менять:

- с помощью ввода цифр;
- клавишами-стрелками;
- кнопками-стрелками в окне редактора.



ПРИМЕЧАНИЕ

Можно использовать совместно с редактором даты ([стр. 33](#)).

Свойства

Значение (Value)

Хранит введенные в окне редактора дату и время, округленное до секунд. Тип значения - уведомляющий timestamp.

При редактировании меняет значение времени, а значение даты оставляет без изменений.

pValueChanged

Ссылка на команду, выполняемую при изменении времени в окне редактора.



ПРИМЕР

При вводе времени в окне редактора проверяется условие: если введенный час вне границ рабочего времени, появляется сообщение "Рабочее время вышло", в противном случае - "Время выбрано". Тогда код обработчика команды:

```
if (DateTime.Hour(TimeEditor.Value) < 9 || DateTime.Hour(TimeEditor.Value) > 18)
    TimeEditor.Error = "Рабочее время вышло";
else TimeEditor.Error = "Время выбрано";
```

Описание ошибки (Error)

Текст ошибки ввода времени. Тип значения - уведомляющий string.



ПРИМЕЧАНИЕ

Свойство не предназначено для проверки введенного времени. Проверку времени можно описать в команде, на которую ссылается свойство **pValueChanged**.

2.1.9.4. Редактор интервала времени

Редактор интервала времени.



Позволяет указать интервал времени с точностью до секунд. Значения полей редактора можно менять:

- с помощью ввода цифр;
- клавишами-стрелками;
- выбрав дату из выпадающего календаря ([стр. 73](#)).

Также можно использовать кнопки смещения к предыдущему и следующему интервалу времени.

Свойства

Begin


Хранит значение, соответствующее началу интервала, указанному в редакторе. Тип значения - уведомляющий timestamp.

End

Хранит значение, соответствующее концу интервала, указанному в редакторе. Тип значения - уведомляющий timestamp.

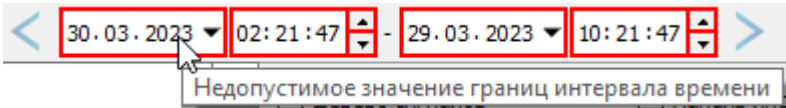
Error

Текст ошибки, возникшей при указании границ интервала. Тип значения - уведомляющий string.



ПРИМЕР

Если указать в качестве начала интервала значение, большее или равное значению окончания интервала, возникнет ошибка.



Текст ошибки записывается в компонент в качестве значения описываемого свойства.

pIntervalChanged

Ссылка на команду, выполняемую при изменении границ интервала.

pShiftCompleted

Ссылка на команду, выполняемую только при изменении границ интервала с помощью кнопок **Перейти к предыдущему интервалу времени**, **Перейти к следующему интервалу времени**.

2.1.9.5. Горизонтальный сплиттер

Компонент, предназначенный для деления области пользовательского интерфейса на левую и правую части с возможностью изменять размеры этих частей с помощью мышки.



Пример построения подобного визуального объекта описан ниже.

1. Поместите объекты, размеры которых должны меняться, в общий "контейнер". В примере на картинке роль таких объектов играют экземпляры типа **Поле ввода**: *Область 1* и *Область 2*, а роль "контейнера" - экземпляр типа **Прямоугольник**. Примените к "контейнеру" инструмент **Линейная разметка** с горизонтальной ориентацией.
2. Добавьте в "контейнер" экземпляр типа **Горизонтальный сплиттер**. Укажите его толщину, начальное положение, направление (в данном случае - «Слева») и границы перемещения - минимум и максимум (подробнее - в описании свойств компонента). Обратите внимание, что в списке **Структура объекта** сплиттер должен находиться между объектами, меняющими размер.

3. К объекту, расположенному слева (*Область 1*), примените инструмент **Ограничения размеров элемента разметки**. Укажите в нем в качестве вычисляемых значений минимальной и максимальной ширины положение горизонтального сплиттера.

Структура объекта

Имя

Описание

▼ MainForm

Тип на основе Форма

▼ Графические объекты

▼ Workspace

Прямоугольник

▼ Графические объекты

▼ LineEdit_1

Поле ввода

▼ Разметка

LayoutSizeLimits

Ограничения размеров элемента разметки

HorizontalSplitter

Горизонтальный сплиттер

LineEdit_2

Поле ввода

▼ Разметка

LinearLayout

Линейная разметка

Редактор свойств

Свойство

Характеристики

Значение

S

Отображаемое имя

LayoutSizeLimits

i8

Кардинальное число

1

▼ f8

Минимальная ширина

R W

10

Начальное значение

<не определено>

Вычисляемое значение

✓

(f) HorizontalSplitter.Position

> f8

Минимальная высота

R W

0

▼ f8

Максимальная ширина

R W

100

Начальное значение

<не определено>

Вычисляемое значение

✓

(f) HorizontalSplitter.Position

> f8

Максимальная высота

R W

16777215

Редактор свойств

События

✓

ПРИМЕЧАНИЕ

Применение горизонтального сплиттера невозможно без экземпляров типов **Линейная разметка** и **Ограничения размеров элемента разметки**.

Свойства

Толщина (Thickness)

Толщина разделителя. Тип значения - double.

Положение (Position)

Положение разделителя относительно левого края "контейнера" по умолчанию. Тип значения - уведомляющий double.

Минимум (Minimum)

Левый ограничитель перемещения сплиттера (относительно левого края "контейнера"). Тип значения - double.

Максимум (Maximum)

Правый ограничитель перемещения сплиттера (относительно левого края "контейнера"). Тип значения - double.

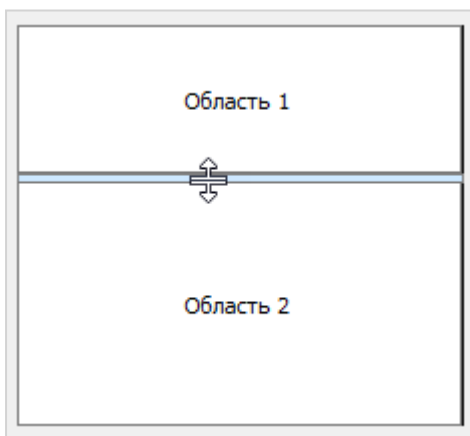
Направление (HorizontalAlignment)

Определяет область "контейнера", ширину которой будет изменять разделитель при перемещении. Тип значения - int4.

Направление "По центру" не используется.

2.1.9.6. Вертикальный сплиттер

Компонент, предназначенный для разделения области пользовательского интерфейса на верхнюю и нижнюю части с возможностью изменять размеры этих частей с помощью мышки.



Пример построения подобного визуального объекта описан ниже.

1. Поместите объекты, размеры которых должны меняться, в общий "контейнер". В примере на картинке роль таких объектов играют экземпляры типа **Поле ввода**: *Область 1* и *Область 2*, а роль "контейнера" - экземпляр типа **Прямоугольник**. Примените к "контейнеру" инструмент **Линейная разметка** с вертикальной ориентацией.
2. Добавьте в "контейнер" экземпляр типа **Вертикальный сплиттер**. Укажите его толщину, начальное положение, направление (в данном случае - «Сверху») и границы перемещения - минимум и максимум (подробнее - в описании свойств компонента). Обратите внимание, что в списке **Структура объекта** сплиттер должен находиться между объектами, меняющими размер.

3. К объекту, расположенному сверху (*Область 1*), примените инструмент **Ограничения размеров элемента разметки**. Укажите в нем в качестве вычисляемых значений минимальной и максимальной высоты положение вертикального сплиттера.

Структура объекта

Имя	Описание
MainForm	Тип на основе Форма
Графические объекты	
Workspace	Прямоугольник
Графические объекты	
LineEdit_1	Поле ввода
Разметка	
LayoutSizeLimits	Ограничения размеров элемента разметки
VerticalSplitter	Вертикальный сплиттер
LineEdit_2	Поле ввода
Разметка	
LinearLayout	Линейная разметка

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		LayoutSizeLimits
Кардинальное число		1
Минимальная ширина	R W	1
Минимальная высота	R W	10
Начальное значение		<не определено>
Вычисляемое значение	✓	{f} VerticalSplitter.Position
Максимальная ширина	R W	16777215
Максимальная высота	R W	50
Начальное значение		<не определено>
Вычисляемое значение	✓	{f} VerticalSplitter.Position

Редактор свойствСобытия

ПРИМЕЧАНИЕ

Применение вертикального сплиттера невозможно без экземпляров типов **Линейная разметка** и **Ограничения размеров элемента разметки**.

Свойства

Толщина (Thickness)

Толщина разделителя. Тип значения - double.

Положение (Position)

Положение разделителя относительно верхнего края "контейнера" по умолчанию. Тип значения - уведомляющий double.

Минимум (Minimum)

Верхний ограничитель перемещения сплиттера (относительно верхнего края "контейнера"). Тип значения - double.

Максимум (Maximum)

Нижний ограничитель перемещения сплиттера (относительно верхнего края "контейнера"). Тип значения - double.

Направление (HorizontalAlignment)

Определяет область "контейнера", ширину которой будет изменять разделитель при перемещении. Тип значения - int4.

Направление "По центру" не используется.

2.1.10. Меню

2.1.10.1. Элемент меню

Элемент контекстного [2.2.1. Меню \(стр. 70\)](#). Используется в меню, но также может использоваться как отдельный компонент.



ПРИМЕЧАНИЕ

После изменения размеров компонента для обновления разметки необходимо вызвать метод **UpdateLayout(keyGestureWidth: double)**. Входной параметр необходимо указать для того, чтобы ширина правой части с сочетаниями клавиш была одинаковой для всех элементов меню.

Свойства

IconFile

Добавляет элементу меню иконку. В качестве значения укажите путь к файлу иконки. Тип значения - уведомляющий string.



ПРИМЕР

"Files_Icons/SaveToFile16.png"

Title

Описание команды меню. Тип значения - уведомляющий string.

KeyGesture

Значение, отображаемое в правой части элемента меню. Тип значения - уведомляющий string.

Checked

Преобразует элемент меню в элемент выбора с иконкой . Тип значения - уведомляющий bool.

Значение:

- «true» - элемент выбора;
- «false» - обычный элемент меню.

Selected

Указывает, выбран (выделен) ли элемент меню. Тип значения - уведомляющий bool.

Значение:

- «true» - элемент выбран;
- «false» - элемент не выбран.

Active

Активность элемента меню. Тип значения - уведомляющий bool.

Значение:

- «true» - элемент активен;
- «false» - элемент неактивен.

Items

Дочерние элементы меню. Тип значения - уведомляющий string.

2.1.10.2. Разделитель элементов меню

Разделитель элементов [2.2.1. Меню \(стр. 70\)](#). Используется в меню, но также может использоваться как отдельный компонент.

Собственных свойств не имеет, так как представляет собой горизонтальную линию.



ПРИМЕЧАНИЕ

После изменения размеров компонента для обновления разметки необходимо вызвать метод **UpdateLayout()**.

2.1.10.3. Результат выбора меню

Компонент, в котором сохраняется результат выбора пункта в [2.2.1. Меню \(стр. 70\)](#).

Экземпляр этого типа не отображается на форме и виден только в области **Структура** объекта.

Свойства

Идентификатор команды (Id)

Идентификатор выбранного (выделенного) пункта меню. Тип значения - string.

Меню остается открытым (StaysOpen)

Указывает, остается ли открытым меню при выборе (выделении) этого пункта меню. Тип значения - bool.

Обновить элементы меню (UpdateItems)

Позволяет обновлять элементы меню после его открытия. Значение данного свойства представляет собой json-строку. Тип значения - уведомляющий string. Подробнее структура json-строки описана в свойстве Items компонента Menu [\(стр. 70\)](#). Строка-значение данного свойства должна описывать измененное состояние пунктов меню.

Используется, например, если выбор одного пункта меню должен приводить к изменению других пунктов меню: их заголовков, иконок, доступности.



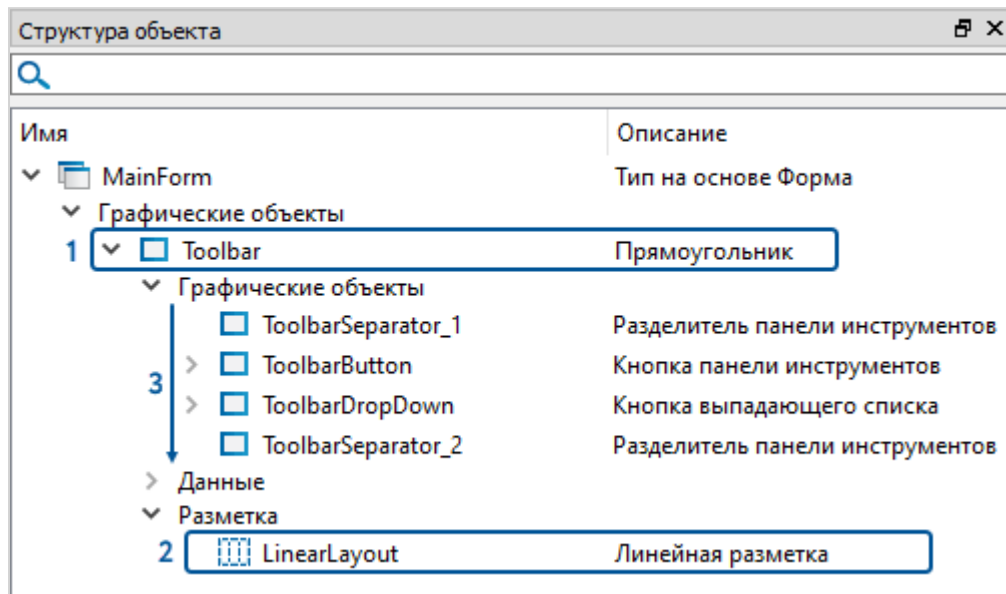
ПРИМЕР

```
MenuResult.UpdateItems = JSON.stringify([
  {
    'id': "addSignal",
    'title': "Добавить новый сигнал...",
    'active': true
  },
  {
    'id': "saveSignal",
    'title': "Сохранить сигнал...",
    'active': true
  }
]);
```

2.1.11. Панель инструментов

Панель инструментов в проекте SePlatform.HMI можно построить из экземпляров компонентов [2.1.11.1. Кнопка панели инструментов \(стр. 44\)](#), [2.1.11.2. Кнопка выпадающего списка \(стр. 46\)](#), [2.1.11.4. Панель поиска \(стр. 48\)](#) и [2.1.11.3. Разделитель панели инструментов \(стр. 47\)](#). Чтобы создать панель инструментов:

1. Создайте графический объект типа **Прямоугольник**.
2. Примените к нему инструмент **Линейная разметка**. Это необходимо для корректного поведения компонентов панели.
3. Экземпляры компонентов панели размещайте внутри созданного прямоугольника. Учитывайте порядок отображения экземпляров в структуре объекта: в таком же порядке они будут расположены на панели инструментов.



Пример построения панели инструментов приведен в [1.4.1. Выбор цвета на панели инструментов из палитры или из меню \(стр. 9\)](#).

2.1.11.1. Кнопка панели инструментов

Кнопка для панели инструментов.



Внешний вид и функции кнопки можно менять с помощью свойств и событий. Кнопку можно использовать совместно с [2.1.11.2. Кнопка выпадающего списка \(стр. 46\)](#). Для визуального разделения панели инструментов можно использовать [2.1.11.3. Разделитель панели инструментов \(стр. 47\)](#).



Для корректного поведения размещайте экземпляр компонента внутри [2.1.11. Панель инструментов \(стр. 43\)](#).

Пример использования приведен в [1.4.1. Выбор цвета на панели инструментов из палитры или из меню \(стр. 9\)](#).

Свойства

IconFile

Путь к файлу иконки для текущего экземпляра кнопки. Тип значения - уведомляющий string.

TitleText

Текст кнопки. Тип значения - уведомляющий string.

Для корректного отображения текста необходимо вручную подобрать ширину кнопки с помощью свойства `ButtonWidth` ([стр. 46](#)).

Checked

Позволяет использовать кнопку как удерживаемую. Тип значения - уведомляющий bool.

Значение:

- «true» - кнопка удерживаемая, выглядит как нажатая.
- «false» - кнопка отпущена, выглядит как обычно.

Active

Позволяет изменить активность кнопки. Тип значения - уведомляющий bool.

Значение:

- «true» - кнопка активна и выглядит как обычно, при наведении мышкой подсвечивается.
- «false» - кнопка неактивна: цвета иконки бледные, при наведении мышкой кнопка не подсвечивается.

IsVisible

Позволяет изменить видимость кнопки. Тип значения - уведомляющий bool.

Значение:

- «true» - кнопка отображается и выглядит как обычно.
- «false» - кнопка не отображается, так как ее ширина становится нулевой, благодаря чему выравниваются остальные элементы панели.

pDropDown

Ссылка на определенный экземпляр [2.1.11.2. Кнопка выпадающего списка \(стр. 46\)](#).

Если указать значение, то в рантайме при наведении мышкой на текущий экземпляр `ToolbarButton` подсвечиваться также будет указанный экземпляр `ToolbarDropDown`.

Margin

Устанавливает размер отступов для расположения иконки относительно кнопки. Тип значения - uint1.

IconSize

Размер иконки для текущего экземпляра кнопки. Тип значения - uint1.

ButtonWidth

Ширина кнопки. Тип значения - uint2.

ButtonHeight

Высота кнопки. Тип значения - uint2.

2.1.11.2. Кнопка выпадающего списка

Кнопка открытия выпадающего элемента ([2.2.1. Меню \(стр. 70\)](#), [2.2.2. Календарь \(стр. 73\)](#), и т.п.) для панели инструментов.



Внешний вид и функции кнопки можно менять с помощью свойств и событий. Кнопку можно использовать совместно с [2.1.11.1. Кнопка панели инструментов \(стр. 44\)](#). Для визуального разделения панели инструментов можно использовать [2.1.11.3. Разделитель панели инструментов \(стр. 47\)](#).



Для корректного поведения размещайте экземпляр компонента внутри [2.1.11. Панель инструментов \(стр. 43\)](#). Пример использования приведен в [1.4.1. Выбор цвета на панели инструментов из палитры или из меню \(стр. 9\)](#).

Свойства

Checked

Позволяет использовать кнопку как удерживаемую. Тип значения - уведомляющий bool.

Значение:

- «true» - кнопка удерживаемая, выглядит как нажатая.
- «false» - кнопка отпущена, выглядит как обычно.

Active

Позволяет изменить активность кнопки. Тип значения - уведомляющий bool.

Значение:

- «true» - кнопка активна и выглядит как обычно, при наведении мышкой подсвечивается.
- «false» - кнопка неактивна: цвет стрелки светло-серый, при наведении мышкой кнопка не подсвечивается.

IsVisible

Позволяет изменить видимость кнопки. Тип значения - уведомляющий bool.

Значение:

- «true» - кнопка отображается и выглядит как обычно.
- «false» - кнопка не отображается, так как ее ширина становится нулевой, благодаря чему выравниваются остальные элементы панели.

pButton

Ссылка на определенный экземпляр [2.1.11.1. Кнопка панели инструментов \(стр. 44\)](#).

Если указать значение, то в рантайме при наведении мышкой на текущий экземпляр **ToolbarDropDown** подсвечиваться также будет указанный экземпляр **ToolbarButton**.

Margin

Устанавливает размер отступов для расположения иконки относительно кнопки. Тип значения - uint1.

ButtonHeight

Высота кнопки. Тип значения - uint2.

2.1.11.3. Разделитель панели инструментов

Разделитель элементов для панелей инструментов.



Для корректного поведения размещайте экземпляр компонента внутри [2.1.11. Панель инструментов \(стр. 43\)](#).

Пример использования приведен в [1.4.1. Выбор цвета на панели инструментов из палитры или из меню \(стр. 9\)](#).

Свойства

IsVisible

Позволяет изменить видимость разделителя. Тип значения - уведомляющий bool.

Значение:

- «true» - разделитель отображается.
- «false» - разделитель не отображается, так как его ширина становится нулевой, благодаря чему выравниваются остальные элементы панели.

ButtonHeight

Высота разделителя. Тип значения - uint2.

2.1.11.4. Панель поиска

Панель поиска.



Представляет собой поле ввода текста для поиска (шаблона) и кнопку запуска процесса поиска.



После ввода шаблона кнопка меняет свое назначение и позволяет очистить результаты поиска.



Свойства

Текст заполнителя (PlaceholderText)

Текст, отображаемый в поле ввода по умолчанию. Тип значения - уведомляющий string.

IsVisible

Меняет видимость панели поиска, меняя ее ширину. Тип значения - уведомляющий bool.

Значение:

- «true» - панель видна, ширина панели обычная;
- «false» - панель скрывается, так как ее высота становится нулевой, благодаря чему выравниваются другие элементы.

Шаблон поиска (Pattern)

Хранит введенный текст для поиска (шаблон). Тип значения - уведомляющий string.

- Изменяется при нажатии клавиши Enter или кнопки запуска процесса поиска.
- Сбрасывается при нажатии клавиши Esc или кнопки очистки результатов поиска.

pPatternChanged

Ссылка на команду, выполняемую при изменении шаблона в [Pattern](#).

2.1.11.5. Тема оформления панели инструментов

Компонент, позволяющий менять тему оформления визуальных компонентов SePlatform.HMI.CommonLib, добавленных на панель инструментов. Свойства компонента, значения которых меняют внешний вид элементов панели инструментов, ссылаются на свойства основного элемента типа [2.1.1. Тема оформления \(стр. 12\)](#) (см. [Свойства \(стр. 49\)](#)).

Экземпляр этого типа не отображается на форме и виден только в области **Структура** объекта.

Свойства

Тема оформления (init_Theme)

Ссылка на экземпляр типа [2.1.1. Тема оформления \(стр. 12\)](#), значения свойств которого должны быть применены на панели инструментов.

Шрифт (Font)

Шрифт основного текста. Ссылается на значение аналогичного свойства экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - string.

Шрифт заголовков (TitleFont)

Шрифт заголовков. Ссылается на значение аналогичного свойства экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - string.

Цвет фона (BackgroundColor)

Цвет основной области компонента. Ссылается на значение свойства [Цвет фона панели инструментов](#) экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет текста (ForegroundColor)

Цвет текста в основной области компонента. Ссылается на значение свойства [Цвет текста на панели инструментов](#) экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет текста при выключении (DisabledFontColor)

Цвет текста неактивных элементов. Ссылается на значение аналогичного свойства экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет фона редактора (EditorBrushColor)

Цвет фона области компонента, в которой можно изменить значение. Ссылается на значение свойства [Цвет редактора на панели инструментов](#) экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет текста в редакторе (EditorFontColor)

Цвет текста в области компонента, в которой можно изменить значение. Ссылается на значение свойства [Цвет текста на панели инструментов](#) экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет выделения (SelectionColor)

Цвет активного (выбранного) элемента в области компонента, в которой можно изменить значение. Ссылается на значение свойства [Цвет выделения на панели инструментов](#) экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет рамки (BorderColor)

Цвет рамки, ограничивающей область компонента, в которой можно изменить значение. Ссылается на значение аналогичного свойства экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет рамки при выключении (DisabledBorderColor)

Цвет рамки, ограничивающей область компонента, в которой можно изменить значение, если эта область неактивна. Ссылается на значение аналогичного свойства экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет фона панели инструментов (ToolbarBrushColor)

Цвет фона панели инструментов. Ссылается на значение аналогичного свойства экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет текста на панели инструментов (ToolbarFontColor)

Цвет текста на панели инструментов. Ссылается на значение аналогичного свойства экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет редактора на панели инструментов (ToolbarEditorColor)

Цвет редактора на панели инструментов. Ссылается на значение аналогичного свойства экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет выделения на панели инструментов (ToolbarSelectionColor)

Цвет выделения элемента на панели инструментов. Ссылается на значение аналогичного свойства экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет фона меню (MenuBrushColor)

Цвет фона меню. Ссылается на значение свойства [Цвет фона панели инструментов](#) экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет текста в меню (MenuFontColor)

Цвет текста в меню. Ссылается на значение свойства [Цвет текста на панели инструментов](#) экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

Цвет текста недоступного элемента меню (MenuDisabledFontColor)

Цвет текста недоступного элемента меню. Ссылается на значение свойства [Цвет текста при выключении](#) экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.

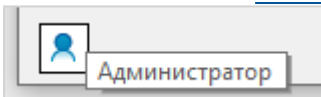
Цвет выделенного элемента меню (MenuSelectionColor)

Цвет выделенного элемента меню. Ссылается на значение свойства [Цвет выделения на панели инструментов](#) экземпляра типа [2.1.1. Тема оформления \(стр. 12\)](#). Тип значения - уведомляющий uint4.



2.1.12. Строка статуса

2.1.12.1. Кнопка авторизации

Кнопка вызова окна [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#).



Внешний вид кнопки зависит от текущего состояния входа:

-  - пользователь авторизован в системе;
-  - гостевой режим.

Во всплывающей подсказке показывается отображаемое имя текущего пользователя.

Свойства

Показывать диалог поверх всех окон (ShowDialogOnTop)

Позволяет настроить отображение открываемого по нажатию на кнопку окна [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#) поверх остальных окон. Тип значения - уведомляющий bool.

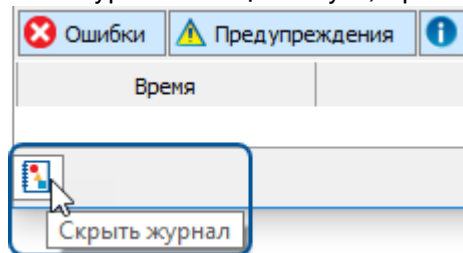
Значение:

- «true» - отображать окно [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#) поверх остальных окон;
- «false» - не отображать окно [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#) поверх остальных окон.

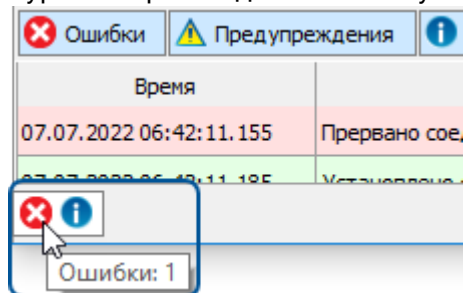
2.1.12.2. Кнопка-индикатор журнала сообщений

Кнопка-индикатор наличия сообщений в журнале ([стр. 66](#)). Позволяет менять видимость журнала.

Если журнал сообщений пуст, в рантайме кнопка выглядит следующим образом:



Если в журнале есть сообщения, на кнопке появляются иконки, соответствующие типам сообщений в журнале. При наведении на иконку всплывает подсказка с количеством сообщений этого типа в журнале:



Свойства

pLogView

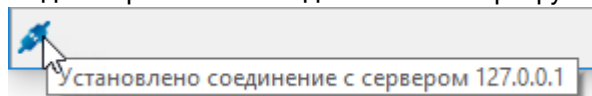
Ссылка на определенный экземпляр журнала сообщений ([стр. 66](#)).

ButtonHeight

Высота кнопки. Тип значения - uint2.

2.1.12.3. Индикатор подключения к серверу

Индикатор состояния подключения к серверу.



Во всплывающей подсказке выводится информация о состоянии подключения.

Свойства

Ссылка на источник (init_ApSource)

Здесь указывается сервер, о состоянии подключения с которым должен сообщать **ConnectionIndicator**. Тип значения - ссылка на источник данных типа **Источник AP**.

Описание состояния (StateInfo)

Содержит информацию о подключении в виде json-строки. Тип значения - уведомляющий string.



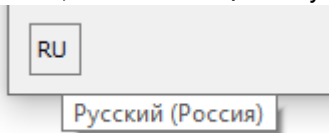
ПРИМЕР

```
{
  "importance": "info",
  "description": "Установлено соединение с сервером 127.0.0.1"
}
```

```
{
  "importance": "error",
  "description": "Прервано соединение с сервером 127.0.0.1"
}
```

2.1.12.4. Кнопка смены раскладки клавиатуры

Кнопка, показывающая текущую раскладку клавиатуры.



При нажатии на кнопку раскладка меняется.



ПРИМЕЧАНИЕ

Кнопка применяется в компонентах [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#) и [2.2.9.2. Диалог смены пароля \(стр. 96\)](#).

2.1.12.5. Кнопка вызова экранной клавиатуры

Кнопка вызова экранной клавиатуры.



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), в котором будут сохранены настройки вызова экранной клавиатуры. Ссылка на созданный экземпляр указывается в свойстве Параметры запуска [\(стр. 54\)](#).

Кнопка уже есть в:

- диалогах авторизации ([2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#), [2.2.9.2. Диалог смены пароля \(стр. 96\)](#));
- диалоге подключения к серверу ([2.2.6. Диалог подключения к серверу \(стр. 80\)](#), [2.2.9.3. Диалог ввода учетных данных \(стр. 97\)](#));
- диалогах ввода значения ([2.2.4. Диалог ввода значения \(стр. 77\)](#));
- диалогах открытия и сохранения файла ([2.2.8.1. Диалог открытия файла \(стр. 83\)](#), [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#)).

Отображение кнопки в указанных компонентах необходимо настроить.

Свойства

Параметры запуска (init_Options)

Ссылка на экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), в котором хранятся настройки вызова экранной клавиатуры.

2.1.12.6. Параметры вызова экранной клавиатуры

Компонент, в котором хранятся настройки вызова экранной клавиатуры.

Экземпляр этого типа не отображается на форме и виден только в области **Структура объекта**.

Свойства

Включено (Enabled)

Настройка видимости кнопки вызова экранной клавиатуры. Тип значения - уведомляющий bool.

Значение:

- «true» - кнопка отображается;
- «false» - кнопка скрыта.

Путь к программе (ProgramPath)

Полный путь к программе экранной клавиатуры. Тип значения - уведомляющий string.

Можно не указывать. По умолчанию будет использоваться:

- в Windows - seplatform.hmi.keyboard.exe (поставляется вместе с дистрибутивом SePlatform.HMI).
- в Linux - /usr/bin/fly-vkbd.

Аргументы запуска (Arguments)

Аргументы запуска программы экранной клавиатуры. Зависят от используемой программы. Тип значения - уведомляющий string.

Аргументы для программы seplatform.hmi.keyboard.exe, поставляемой вместе с дистрибутивом SePlatform.HMI для Windows, приведены в таблице ниже.

Аргумент	Назначение
--window-pos-x VALUE, --window-pos-y VALUE	Управляет стартовой позицией окна, где VALUE - координата верхнего левого угла окна.
--window-height VALUE, --window-width VALUE	Управляет стартовыми размерами окна, где VALUE - высота/ширина окна.

Аргумент	Назначение
--no-window-resize	Запрещает изменять размер окна экранной клавиатуры.
--no-window-minimize	Запрещает сворачивать окно экранной клавиатуры.
--no-window-maximize	Запрещает разворачивать окно экранной клавиатуры.
--no-window-close	Запрещает закрывать окно экранной клавиатуры.
--use-security	<p>Учитывает право доступа к настройкам экранной клавиатуры текущего пользователя подсистемы безопасности.</p> <ul style="list-style-type: none"> ➤ Если в SePlatform.Security пользователю назначено право EditSettings (Изменение настроек) со значением «true», он сможет изменить настройки экранной клавиатуры по нажатию кнопки Настройки. ➤ Если право назначено со значением «false» или не назначено вовсе, кнопка Настройки для него будет заблокирована.

2.1.13. Диалоги

2.1.13.1. Результат выбора папки

Компонент, в котором сохраняется путь к папке, выбранной в окне [2.2.8.3. Диалог выбора папки \(стр. 93\)](#).

Экземпляр этого типа не отображается на форме и виден только в области **Структура** объекта.

Свойства

Путь к папке (FolderPath)

Хранит путь к папке, выбранной в окне [2.2.8.3. Диалог выбора папки \(стр. 93\)](#). Тип значения - уведомляющий string.

2.1.13.2. Результат выбора цвета

Компонент, в котором сохраняется значение, введенное в окне [2.2.5. Диалог выбора цвета \(стр. 78\)](#).

Экземпляр этого типа не отображается на форме и виден только в области **Структура** объекта.

Свойства

Цвет (Color)

Хранит цвет, выбранный в окне [2.2.5. Диалог выбора цвета \(стр. 78\)](#). Тип значения - уведомляющий uint4.

Пользовательская палитра (CustomColors)

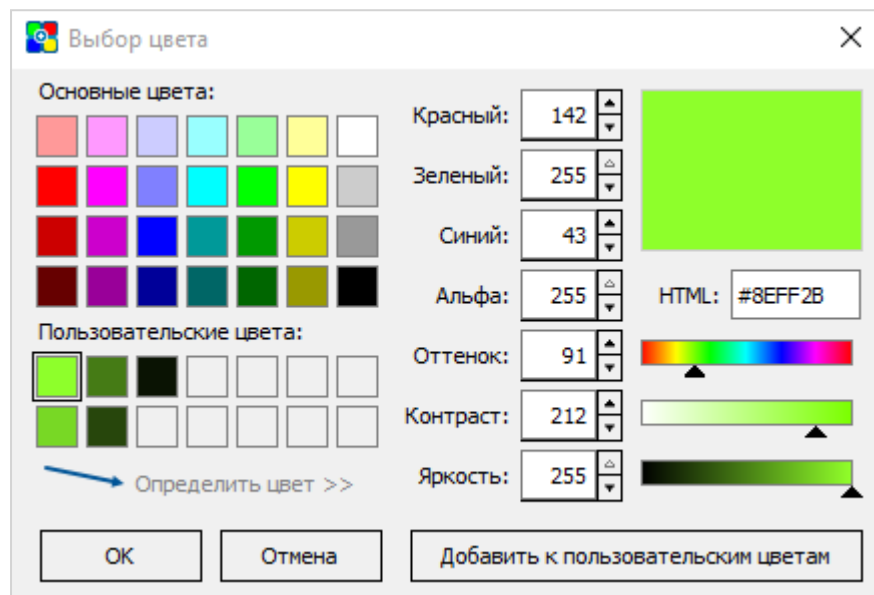
Хранит перечень пользовательских цветов в виде массива чисел в формате JSON. Тип значения - string.



ПРИМЕР

```
[ "0xfffffa500", "0xff9400d3" ]
```

Добавить цвет в список пользовательских можно, расширив диалоговое окно и нажав кнопку **Добавить к пользовательским цветам**.



Всего может быть указано до 14 цветов.

2.1.13.3. Результат выбора параметров подключения

Компонент, в котором сохраняются адрес и порт сервера, к которому выполняется подключение в окне [2.2.6. Диалог подключения к серверу \(стр. 80\)](#).

Экземпляр этого типа не отображается на форме и виден только в области **Структура объекта**.

Свойства

Название (Title)

Название подключения. Тип значения - string.

Хост (Location)

IP-адрес или имя сервера (компьютера), к которому выполняется подключение. Тип значения - string.

Порт (Port)

Номер порта, через который выполняется подключение к указанному серверу (компьютеру). Тип значения - uint2.

Порт истории (HistoryPort)

Номер порта истории. Тип значения - uint2.

Учетные данные (Credentials)

Учетные данные, указываемые при подключении к защищенному паролем серверу. Тип значения - string.

2.1.13.4. Результат выбора даты и времени

Компонент, в котором сохраняется дата, выбранная в [2.2.2. Календарь \(стр. 73\)](#).

Экземпляр этого типа не отображается на форме и виден только в области **Структура** объекта.

Свойства

Значение (Value)

Хранит дату, выбранную в [2.2.2. Календарь \(стр. 73\)](#). Тип значения - timestamp.

2.1.13.5. Результат ввода значения

Компонент, в котором сохраняется значение, введенное в окне [2.2.4. Диалог ввода значения \(стр. 77\)](#).

Экземпляр этого типа не отображается на форме и виден только в области **Структура** объекта.

Свойства

Значение (Value)

Итоговое значение, введенное пользователем. Тип значения - уведомляющий string.

Вводимый текст (InputText)

Вводимый текст. Тип значения - уведомляющий string.

Может использоваться:

- для проверки введенного текста;
- в качестве начального значения при открытии окна **InputDialog** ([стр. 77](#)).

Описание ошибки (Error)

Описание ошибки ввода. Тип значения - уведомляющий string.

При возникновении ошибки ввода:

- вокруг поля ввода появляется красная рамка;
- во всплывающей подсказке появляется текст ошибки;
- кнопка **OK** блокируется.

ПРИМЕР

Допустим, значение в поле ввода не должно быть пустым.
Проверка условия может быть описана на языке SePlatform.Ом компактной тернарной операцией вида:

```
<condition>?<statement1>:<statement2>
```

Графические объекты

Данные

FolderName

CreateFolder

OkButton

CancelButton

Данные

Функции

Результат ввода значения

Команда

Кнопка

Кнопка

Редактор свойств

Свойство

Характеристики

Значение

Отображаемое имя

Кардинальное число

Значение

Вводимый текст

Описание ошибки

Вычисляемое значение

FolderName

1

R W f

R W f

R W f

(f) InputText == "" ? "Имя папки не должно быть пустым" : ""

В данном случае проверяется вводимый текст (**InputText**). При пустом значении ввода:

➤ вокруг поля ввода появится красная рамка;

➤ во всплывающей подсказке появится текст ошибки;

➤ кнопка **OK** заблокируется.

Имя новой папки

Имя папки не должно быть пустым

OK

Отмена

2.1.13.6. Результат редактирования колонтитула

Компонент, в который сохраняется результат редактирования колонтитула в окне [2.2.7. Редактор колонтитула \(стр. 82\)](#) – текст и выравнивание. Чтобы использовать результат, обратитесь к свойствам компонента.

Экземпляр этого типа не отображается на форме и виден только в области **Структура** объекта.

Свойства

Text

Текст колонтитула. Тип значения – уведомляющий string.

TextAlignment

Выравнивание текста колонтитула. Тип значения – уведомляющий int4.

2.1.13.7. Результат выбора файла

Компонент, в котором сохраняется имя и тип файла, выбранного в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) или [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#).

Экземпляр этого типа не отображается на форме и виден только в области **Структура** объекта.

Свойства

Путь к папке (FolderPath)

Хранит полный путь к папке, в которой находится файл, выбранный в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) или [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#). Тип значения – уведомляющий string.

Имя файла (FileName)

Хранит имя файла, выбранного в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) или [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#). Тип значения – уведомляющий string.

Путь к файлу (FilePath)

Хранит полный путь к файлу, выбранному в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) или [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#). Тип значения – уведомляющий string.

Тип файла (FileType)

Хранит индекс типа файла, выбранного в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) или [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#). Тип значения – уведомляющий int4.

Индексом типа файла является порядковый номер типа в списке фильтра типа файлов. Список фильтра создается в соответствующем свойстве - **FileTypeFilter (Фильтр типа файлов)** - компонентов Диалог открытия файла ([стр. 84](#)) или Диалог сохранения файла ([стр. 89](#)).

2.1.13.8. Результат диалога авторизации

Компонент, через который передаются id и отображаемое имя текущего пользователя от формы [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#) к форме [2.2.9.2. Диалог смены пароля \(стр. 96\)](#). Пример применения приведен в описании компонента [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#).

Экземпляр этого типа не отображается на форме и виден только в области **Структура объекта**.

Свойства

UserId

Хранит уникальный идентификатор (uid) текущего пользователя подсистемы безопасности SePlatform.Security. Тип значения - уведомляющий string.



ПРИМЕР

Вид значения: alpha:xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx.

Пример значения: «a1pha:c3a997b1-326e-42ee-8907-e359fc17feb7».

UserDisplayName

Хранит отображаемое имя текущего пользователя подсистемы безопасности SePlatform.Security. Тип значения - уведомляющий string.



ПРИМЕР

Пример значения: «Иванов».

2.1.13.9. Список файлов

Список файлов и папок, содержащихся в указанной папке.

Имя	Дата изменения	Размер	↑
configuration_files	17.11.2022 14:12:29		
new_backups	17.11.2022 14:12:29		
old_backups	17.11.2022 14:12:29		
17112022_bckp.csv	17.11.2022 14:12:53	6377	
17112022_bckp.pdf	17.11.2022 14:12:53	6377	
18112022_bckp.csv	17.11.2022 14:12:53	6377	
18112022_bckp.pdf	17.11.2022 14:12:53	6377	↓

Указать папку, содержимое которой будет отображаться в компоненте, настроить фильтры отображения файлов и настроить ограничения на просмотр файловой системы можно с помощью свойств компонента.

Свойства

Корневой путь (RootPath)

Ограничивает область просмотра файловой системы. Тип значения - уведомляющий string.

Отображаемая папка (BrowsedPath)

Путь к папке, содержимое которой должно отображаться в окне [2.1.13.9. Список файлов \(стр. 60\)](#). Тип значения - уведомляющий string.

Полнота отображения - только папки, папки и все файлы или папки и файлы определенного типа - указывается в свойстве [ExtensionFilter \(Фильтр по типу файла\)](#).

Фильтр по типу файла (ExtensionFilter)

Фильтр типа файлов, отображаемых в окне [2.1.13.9. Список файлов \(стр. 60\)](#). Тип значения - уведомляющий string.

- Если не указывать значение, то в списке будут отображаться все файлы и папки.
- Если указать конкретное расширение, например «*.csv», то будут отображаться папки и файлы с указанным расширением.

Показывать свойства файлов (ShowFileProperties)

Позволяет отображать больше информации о файлах и папках в окне [2.1.13.9. Список файлов \(стр. 60\)](#). Тип значения - уведомляющий bool.

Значение:

- «true» - для файлов и папок отображаются имена и даты изменений, а для файлов - еще размеры (в байтах);

Имя	Дата изменения	Размер
new_backups	17.11.2022 14:12:29	
old_backups	17.11.2022 14:12:29	
17112022_bckp.csv	17.11.2022 14:12:53	6377
18112022_bckp.csv	17.11.2022 14:12:53	6377
19112022_bckp.csv	17.11.2022 14:12:53	6377

- «false» - отображаются только имена файлов и папок.

new_backups
old_backups
17112022_bckp.csv
18112022_bckp.csv
19112022_bckp.csv

Разрешена смена папки (ChangeFolderAllowed)

Указывает на возможность смены папки, открывающейся в окне [2.1.13.9. Список файлов \(стр. 60\)](#). Тип значения - уведомляющий bool.

Значение:

- «true» - можно сменить открывшуюся папку;
- «false» - нельзя сменить открывшуюся папку.

Выделенный путь (SelectedPath)

Путь к файлу или папке, выбранной в окне [2.1.13.9. Список файлов \(стр. 60\)](#). Тип значения - уведомляющий string.



ПРИМЕР

Допустим, в свойстве **BrowsedPath (Отображаемая папка)** указано значение `D:\CommonLib\backups`, тогда в окне **FileList** отображается содержимое этой папки.

Имя	Дата изменения	Размер
configuration_files	17.11.2022 14:12:29	
new_backups	17.11.2022 14:12:29	
old_backups	17.11.2022 14:12:29	
17112022_bckp.csv	17.11.2022 14:12:53	6377
18112022_bckp.csv	17.11.2022 14:12:53	6377

В таком случае свойство **SelectedPath (Выделенный путь)**:

- при выборе папки new_backups примет значение «`D:\CommonLib\backups\new_backups`»;
- при выборе файла 17112022_bckp.csv примет значение «`D:\CommonLib\backups\17112022_bckp.csv`».

Выделенный файл (SelectedFileName)

Название файла, выбранного в окне [2.1.13.9. Список файлов \(стр. 60\)](#). Тип значения - уведомляющий string.



ПРИМЕР

Допустим, в свойстве **BrowsedPath (Отображаемая папка)** указано значение `D:\CommonLib\backups`, тогда в окне **FileList** отображается содержимое этой папки.

Имя	Дата изменения	Размер
configuration_files	17.11.2022 14:12:29	
new_backups	17.11.2022 14:12:29	
old_backups	17.11.2022 14:12:29	
17112022_bckp.csv	17.11.2022 14:12:53	6377
18112022_bckp.csv	17.11.2022 14:12:53	6377

В таком случае:

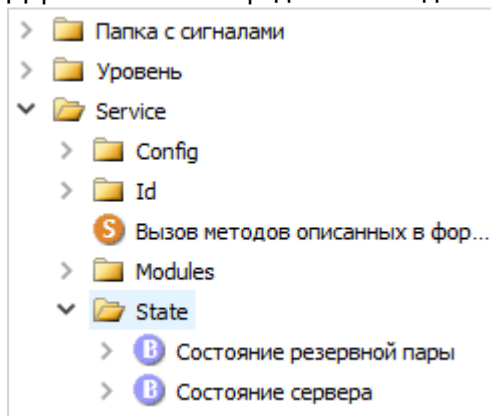
- при выборе файла `17112022_bckp.csv` свойство **SelectedFileName (Выделенный файл)** примет значение «`17112022_bckp.csv`»;
- при выборе папки:
 - если ранее не был выбран ни один файл, то значение свойства **SelectedFileName (Выделенный файл)** останется пустым;
 - если ранее был выбран какой-либо файл, то имя этого файла будет храниться в качестве значения свойства **SelectedFileName (Выделенный файл)**, пока не будет выбран другой файл и значение не перезапишется.

OnPathChosen

Ссылка на команду, выполняемую в результате двойного щелчка ЛКМ по строке файла в списке.

2.1.14. Дерево сигналов

Дерево сигналов. Предназначено для отображения адресного пространства сервера в виде дерева.



2.1.14.1. Свойства

Вывод описания предпочтительнее имени (DescriptionPreferred)

Позволяет выбрать, что отображать в дереве: имя сигнала или его описание. Тип значения - уведомляющий bool.

Значение:

- «true» - отображать описание сигнала (если не указано, то имя);
- «false» - отображать имя сигнала.

Ссылка на источник (init_ApSource)

Здесь указывается сервер, дерево сигналов которого должно отображаться в **ItemTree**. Тип значения - ссылка на источник данных типа **Источник АР**.

pDragStarted

Ссылка на экземпляр типа [2.1.7. Обработчик события \(стр. 28\)](#), активируемый при перетаскивании узла дерева левой кнопкой мыши.

pDoubleClicked

Ссылка на экземпляр типа [2.1.7. Обработчик события \(стр. 28\)](#), активируемый при двойном клике левой кнопкой мыши по узлу дерева.

pMouseRightClicked

Ссылка на команду, выполняемую при клике правой кнопкой мыши по узлу дерева.

Может использоваться, например, для вызова контекстного меню [\(стр. 70\)](#).

2.1.14.2. Функции

Find

```
void Find(string pattern)
```

Выполняет поиск сигналов в дереве по шаблону названия, указанному в качестве значения входного параметра `pattern`.

Поиск выполняется на стороне сервера. Результат поиска отображается в дереве.

CancelFind

```
void CancelFind()
```

Отменяет поиск сигналов, выполняемый функцией **Find()**.

После выполнения этой функции в дереве сигналов вместо результатов поиска отображаются все сигналы сервера.

Функция не требует входных параметров.

RefreshCurrentItem

```
void RefreshCurrentItem()
```

Заново запрашивает список дочерних узлов для узла, выделенного в дереве.

Функция не требует входных параметров.

GetCurrentItemInfo

```
string GetCurrentItemInfo()
```

Возвращает описание выделенного в дереве сигнала в виде json-строки.

Функция не требует входных параметров.



ПРИМЕР

Пример возвращаемого результата:

```
{
    "tag": "Level.s_level",
    "dataType": "int1",
    "icon": "CommonLib_Icons/Items/SByte.png"
}
```

Пример использования в обработке команды, на которую ссылается pDragStarted ([стр. 64](#)):

```
// получение значения, возвращаемого функцией
var json = GetCurrentItemInfo();
if (json != "")
{
    // разбираем объект info
    var info = JSON.parse(json);

    // записываем в журнал сообщения о начале перетаскивания сигнала с именем
    <tag>
    LogView.AddMessage(0, "Начато перетаскивание " + info.tag);
}
```

2.1.15. Журнал сообщений

Журнал сообщений.

<div> <div>❌ Ошибки</div> <div>⚠ Предупреждения</div> <div>ℹ Информация</div> <div>⬇</div> <div>📄</div> <div>Поиск в описании</div> <div>🔍</div> </div>	
Время	Описание
06.07.2022 12:53:40.214	Прервано соединение с сервером 127.0.0.1
06.07.2022 12:53:40.217	Плохое соединение с сервером 127.0.0.1
06.07.2022 12:53:40.220	Соединение с сервером 127.0.0.1 установлено
06.07.2022 12:53:40.223	Выполняется соединение с сервером 127.0.0.1

Отображает сообщения в виде таблицы. По умолчанию в таблице отображаются только **Время** и **Описание** сообщений. Чтобы выводить дополнительную информацию, измените значение свойства `AdditionalColumns` ([стр. 67](#)).

Сообщения в журнале могут иметь разную важность, обозначаемую цветом фона сообщения:

- ошибки - красные;
- предупреждения - желтые;
- информационные сообщения: обычные - зеленые, и отладочные - белые.

Подсказки по работе с журналом, добавленным в проект, в режиме исполнения:

- чтобы в журнале не отображались сообщения определенного типа, следует отжать кнопку с названием типа на панели инструментов журнала;
- чтобы следить за новыми сообщениями, нужно нажать кнопку **Автоматическое прокручивание списка**;
- чтобы очистить журнал сообщений, нажмите кнопку **Очистить список**;
- чтобы найти в журнале сообщения с определенной фразой, введите ее в поле **Поиск в описании** и нажмите кнопку **Поиск** 🔍.

2.1.15.1. Свойства

IsVisible

Меняет видимость журнала. Тип значения - уведомляющий bool.

Значение:

- «true» - показать журнал;
- «false» - скрыть журнал.

Еще изменить видимость журнала можно при помощи [2.1.12.2. Кнопка-индикатор журнала сообщений \(стр. 51\)](#).

Количество ошибок (ErrorCount)

Хранит текущее количество сообщений об ошибках в журнале. Тип значения - уведомляющий int4.

Количество предупреждений (WarningCount)

Хранит текущее количество предупреждений в журнале. Тип значения - уведомляющий int4.

Количество сообщений (InformationCount)

Хранит текущее количество информационных сообщений в журнале. Тип значения - уведомляющий int4.

Дополнительные столбцы (AdditionalColumns)

Перечень дополнительных столбцов таблицы сообщений в виде json-строки. Тип значения - string.



ПРИМЕР

Пример значения свойства:

```
[
  { "id": "address", "title": "Адрес сервера" },
  { "id": "port", "title": "Номер порта" }
]
```

Результат в рантайме - новые столбцы Адрес сервера и Номер порта:

<div> ❌ Ошибки ⚠️ Предупреждения ℹ️ Информация 📶 🔄 </div> <div>Поиск в описании 🔍</div>			
Время	Описание	Адрес сервера	Номер порта
06.07.2022 12:58:40.071	Установлено соединение с сервером 127.0.0.1	127.0.0.1	4388
06.07.2022 12:58:40.082	Выполняется подключение к серверу 127.0.0.1	127.0.0.1	4388

2.1.15.2. Функции

AddMessage

```
void AddMessage(int4 importance, string text)
```

Добавляет в журнал сообщение с заданной важностью. Если в свойстве AdditionalColumns ([стр. 67](#)) указаны дополнительные столбцы, используйте функцию **Add()**.

Входные параметры:

- важность;
- текст сообщения.

Важность обозначается цифрой:

- «0» - отладочное информационное сообщение, цвет фона - белый;
- «1» - информационное сообщение, цвет фона - зеленый;
- «2» - важное сообщение, цвет фона - желтый;
- «3» - сообщение об ошибке, цвет фона - красный.



ПРИМЕР

```
LogView.AddMessage(3, "Прервано соединение с сервером 127.0.0.1")
```

Add

```
void Add(int4 importance, string text, string additional)
```

Добавляет в журнал сообщение с заданной важностью и дополнительными данными. Эта функция используется, если в свойстве `AdditionalColumns` ([стр. 67](#)) указаны дополнительные столбцы.

Входные параметры:

- важность;
- текст сообщения;
- дополнительные данные в виде json-строки:
{ "id1":value1, "id2":value2, ... },
помещаемые в дополнительные столбцы таблицы сообщений.



ПРИМЕР

При вызове функции в JavaScript понадобится метод **JSON.stringify()** для преобразования данных в json-строку:

```
Log.Add(3, "Прервано соединение с сервером 127.0.0.1", JSON.stringify  
({"address":"127.0.0.1", "port":4388}))
```

При вызове функции в SePlatform.Ом представьте данные в виде строки, заключив их в кавычки, а кавычки внутри строки экранируйте обратным слешем:

```
Log.Add(3, "Прервано соединение с сервером 127.0.0.1", "{  
\"address\": \"127.0.0.1\", \"port\":4388 }")
```

Clear

```
void Clear()
```

Очищает журнал сообщений.

GetViewSettings

```
string GetViewSettings()
```

Возвращает настройки внешнего вида журнала в виде json-строки.

Настройки внешнего вида содержат информацию о ширине столбцов, порядке сортировки, включенных кнопках фильтрации, текста поиска, режим слежения за последним сообщением. Эти настройки могут использоваться для восстановления внешнего вида журнала с помощью функции [UpdateView\(\)](#) при перезапуске проекта.

**ПРИМЕР**

Пример возвращаемого значения:

```
{
  "Columns": {
    "importance": {
      "Index": 0,
      "Width": 20,
      "Hidden": true
    },
    "time": {
      "Index": 1,
      "Width": 150,
      "Hidden": false
    },
    "text": {
      "Index": 2,
      "Width": 424,
      "Hidden": false
    },
    "address": {
      "Index": 3,
      "Width": 100,
      "Hidden": false
    },
    "port": {
      "Index": 4,
      "Width": 100,
      "Hidden": false
    }
  },
  "Filter": {
    {
      "Errors": true,
      "Warnings": true,
      "Informations": true,
      "Text": ""
    },
    "FollowLatest": false
  }
}
```

UpdateView

```
void UpdateView(string settings)
```

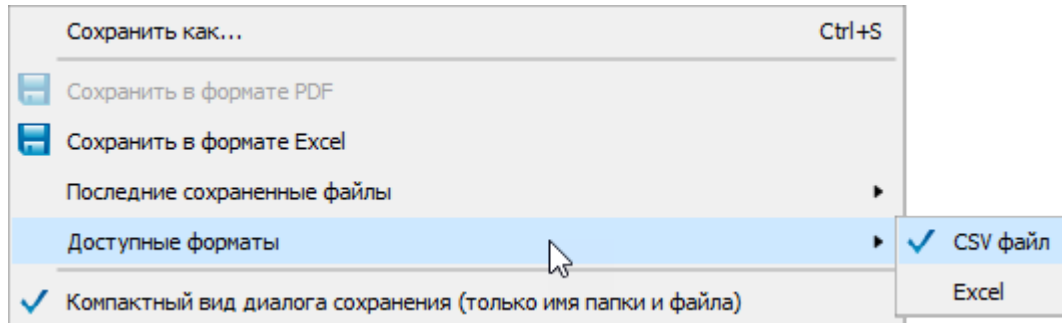
Настраивает внешний вид журнала в соответствии с настройками, которые могут быть получены в результате вызова функции [GetViewSettings\(\)](#).

Входной параметр - настройки внешнего вида журнала в виде json-строки.

2.2. Экранные формы

2.2.1. Меню

Контекстное меню.



Меню следует настроить. Перечень элементов меню укажите в качестве значения свойства `Items` ([стр. 70](#)) в виде JSON массива.



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа [2.1.10.3. Результат выбора меню \(стр. 43\)](#), в котором будет сохранен результат выбора элемента меню. Ссылка на созданный экземпляр указывается в свойстве `Ссылка на результат` ([стр. 73](#)).

Пример использования приведен в [1.4.1. Выбор цвета на панели инструментов из палитры или из меню \(стр. 9\)](#).

2.2.1.1. Свойства

PointHeight

Высота элемента, относительно которого открывается меню. Тип значения - `int4`.

IsSubmenu


Указывает, является ли данный экземпляр `Menu` подменю другого экземпляра `Menu`. Тип значения - `bool`.

Элементы меню (Items)

Хранит перечень элементов меню в виде JSON массива, содержащего описание каждого элемента. Тип значения - уведомляющий `string`.

Структура массива следующая:

- идентификатор команды `id` и заголовок `title` - обязательные параметры: `{ 'id': "cmd", 'title': "Команда" }`;

- параметр `checked` со значением «true» - создаст элемент выбора с иконкой : «{ 'id': "slim", 'title': "Компактный вид", 'checked': true }»;
- параметр `icon` добавит элементу меню иконку, находящуюся по указанному пути: «{ 'id': "save", 'title': "Сохранить", 'icon': "Files_Icons/SaveToFile16.png" }»;
- параметр `staysOpen` со значением «true» - оставляет окно меню открытым при выборе элемента: «{ 'id': i, 'title': "Сохранить как...", 'staysOpen': true }»;
- параметр `keyGesture` добавит текст в правой части элемента меню. Этот параметр обычно используют, чтобы указать текстовое описание сочетания "горячих" клавиш: «{ 'id': i, 'title': "Сохранить как...", 'keyGesture': "Ctrl+S" }». Обратите внимание, что выполнение самой команды по нажатию "горячих" клавиш необходимо настроить в ОС;
- разделитель элементов меню задается в виде значения параметра `title`: «{ 'title': "-" }».

ПРИМЕР

Создана функция **GetItems()**. JSON массив пунктов меню описан в функции на языке JavaScript.

```
var items = [
  {
    'id': "csv",
    'title': "Открыть файл CSV",
    'keyGesture': "Ctrl+O"
  },
  {
    'id': "xlsx",
    'title': "Открыть файл Excel",
    'keyGesture': "Ctrl+Shift+O"
  }
];
return JSON.stringify(items);
```

Тогда в качестве значения свойства указана ссылка на функцию.

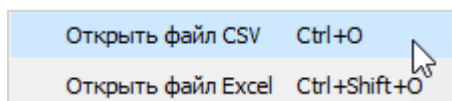
The screenshot shows two windows from an IDE:

- Структура объекта (Object Structure):** A tree view showing the project structure. Under 'MainForm' > 'Графические объекты' (Graphic Objects) > 'ShowMenu_Button' > 'Данные' (Data), there is a 'showMenu' property of type 'Команда' (Command). Under 'Функции' (Functions), there is a 'GetItems' function.
- События (Events):** A table showing event handlers for the 'Invoker' object. The table has columns: 'Имя' (Name), 'Характеристики' (Characteristics), and 'Обработчик' (Handler).

Имя	Характеристики	Обработчик
Invoker		Открыть <Menu> в новом окне
Handler_1		
PointHeight	R W	<не определено>
IsSubmenu	R W	<не определено>
Элементы меню	R W ✓	(f) GetItems()
Ссылка на результат	R ↗ ✓	MenuResult
pAction	R ↗ ✓	OnComplete
pPrevMenu	R ↗	<не определено>
NextMenu	R ↗	<не определено>

A blue arrow points from the 'GetItems' function in the 'Functions' list to the 'Элементы меню' (Menu Items) event handler in the 'Events' list.

Приведенный в примере JSON массив в проекте в режиме исполнения выглядит так:



Ссылка на результат (pResult)

Ссылка на экземпляр типа [2.1.10.3. Результат выбора меню \(стр. 43\)](#), в котором будет сохранен результат выбора пункта меню.

pAction

Ссылка на команду, выполняемую при выборе элемента меню.

pPrevMenu

Ссылка на команду, выполняемую при переходе к предыдущему пункту меню (при управлении меню клавишами).

pNextMenu

Ссылка на команду, выполняемую при переходе к следующему пункту меню (при управлении меню клавишами).

pOnDoClose

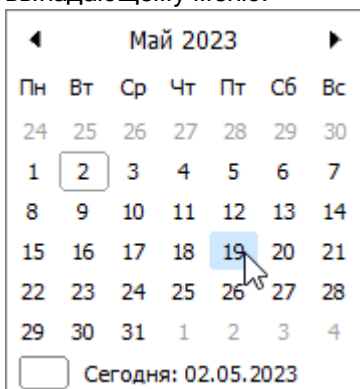
Вспомогательная ссылка на команду, выполняемую перед закрытием меню. Используется для передачи команды закрытия основного меню от подменю.

pOnClosed

Ссылка на команду, выполняемую при закрытии меню.

2.2.2. Календарь

Календарь предназначен для выбора даты. Он используется в редакторе выбора даты аналогично выпадающему меню.



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа [2.1.13.4. Результат выбора даты и времени \(стр. 57\)](#), в котором будет сохранен результат выбора даты. Ссылка на созданный экземпляр указывается в свойстве Ссылка на результат ([стр. 74](#)).

2.2.2.1. Свойства

PointHeight

Высота элемента, относительно которого открывается календарь. Тип значения - int4.

Дата (Date)

Дата, выделенная при открытии окна [2.2.2. Календарь \(стр. 73\)](#). Тип значения - timestamp.

Ссылка на результат (pResult)

Ссылка на экземпляр типа [2.1.13.4. Результат выбора даты и времени \(стр. 57\)](#), в котором будет сохранена выбранная дата.

pSelected

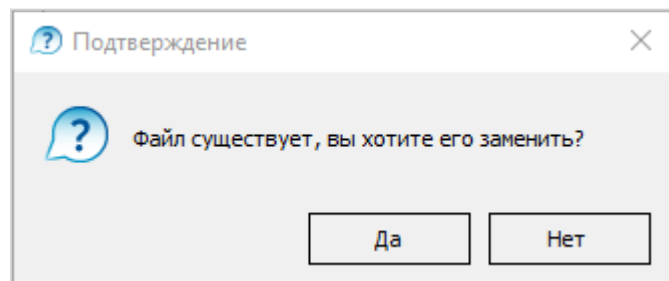
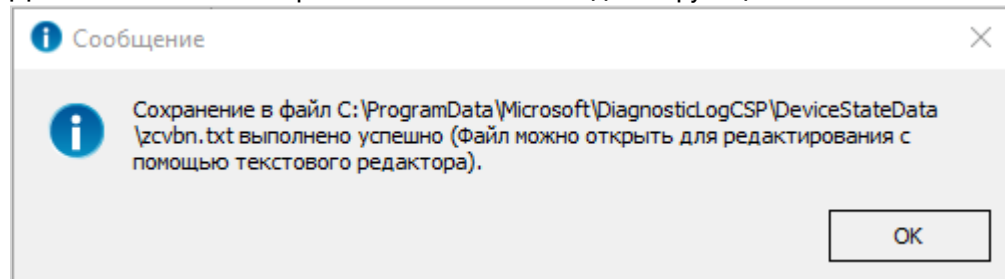
Ссылка на команду, выполняемую при выборе даты в окне [2.2.2. Календарь \(стр. 73\)](#).

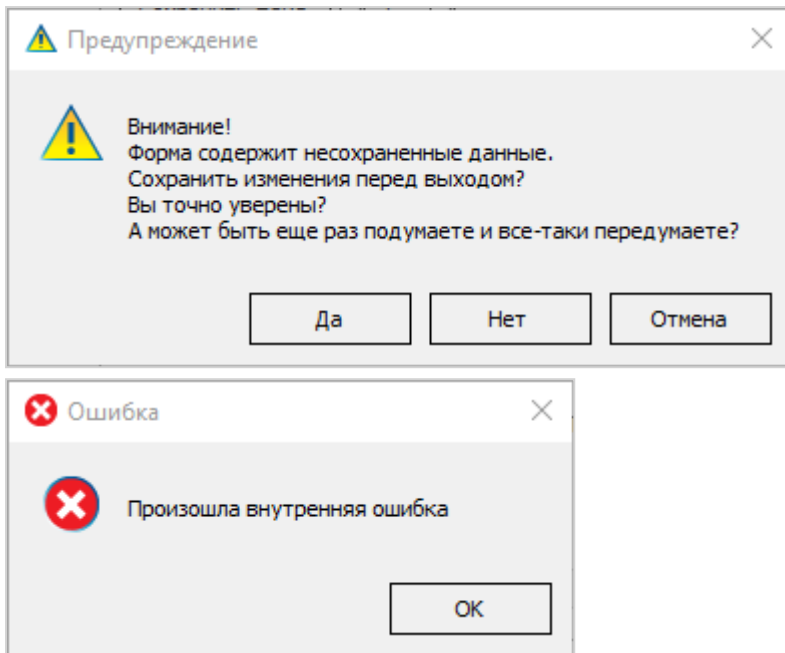
pOnClosed

Ссылка на команду, выполняемую при закрытии окна [2.2.2. Календарь \(стр. 73\)](#).

2.2.3. Сообщение

Диалоговое окно с настраиваемым внешним видом и функциями.



**ПРИМЕЧАНИЕ**

Для изменения внешнего вида и функций диалогового окна, используйте свойства.

2.2.3.1. Свойства

Высота строки текста (LineHeight)

Высота строки текста сообщения. Тип значения - double.

Используется для изменения размера диалогового окна.

Символов в строке (CharactersPerLine)

Количество символов в строке. Тип значения - int4.

Используется для изменения размера диалогового окна.

Сообщение (Message)

Текст сообщения. Тип значения - string.

Иконка (Icon)

Иконка рядом с текстом сообщения. Тип значения - уведомляющий int4.

Может принимать только одно из допустимых значений, приведенных в таблице.

Допустимое значение	Имя значения	Результат выбора значения
«0»	Info	
«1»	Question	
«2»	Warning	
«3»	Error	

Кнопки (Buttons)

Набор кнопок в диалоговом окне. Тип значения - уведомляющий int4.

Может принимать только одно из допустимых значений, приведенных в таблице.

Допустимое значение	Набор кнопок
«0»	ОК
«1»	ОК и Отмена
«2»	Да и Нет
«3»	Да, Нет и Отмена

OkAction

Ссылка на команду, выполняемую при нажатии на кнопку **ОК**.

YesAction

Ссылка на команду, выполняемую при нажатии на кнопку **Да**.

NoAction

Ссылка на команду, выполняемую при нажатии на кнопку **Нет**.

CancelAction

Ссылка на команду, выполняемую при нажатии на кнопку **Отмена**.

Результат (Result)

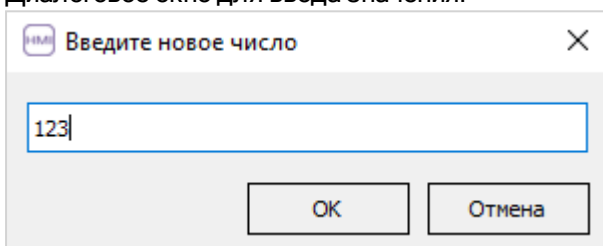
Хранит значение кнопки, выбранной в диалоговом окне. Тип значения - уведомляющий int4.

Значения кнопок приведены в таблице.

Выбранная кнопка	Значение свойства
ОК	«1»
Да	«2»
Нет	«3»
Отмена	«4»

2.2.4. Диалог ввода значения

Диалоговое окно для ввода значения.



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа [2.1.13.5. Результат ввода значения \(стр. 57\)](#), в котором будет сохранено имя и тип выбранного файла. Ссылка на созданный экземпляр указывается в свойстве Ссылка на результат ([стр. 77](#)).

2.2.4.1. Свойства

Ссылка на результат (Result)

Ссылка на [2.1.13.5. Результат ввода значения \(стр. 57\)](#), в котором будет сохранен ввод пользователя.



ПРИМЕЧАНИЕ

Начальное значение в окне **InputDialog** можно настроить. Для этого:

- укажите в качестве значения свойства **Result** компонента **InputDialog** экземпляр типа **InputDialogResult**;
- у указанного экземпляра **InputDialogResult** укажите начальное значение в качестве значения свойства **InputText**.

OkAction

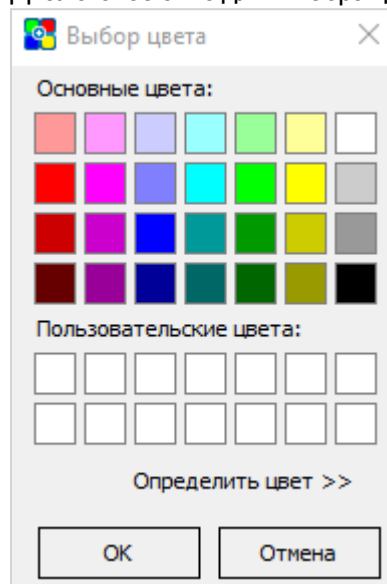
Ссылка на команду, выполняемую при нажатии на кнопку **ОК**.

Вызов экранной клавиатуры (init_ScreenKeyboard)

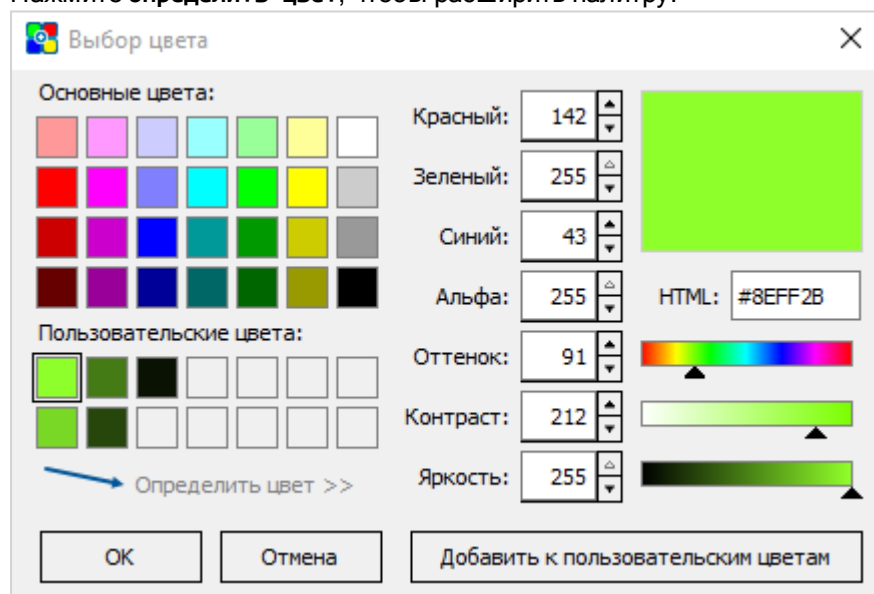
Ссылка на экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), хранящий настройки вызова экранной клавиатуры.

2.2.5. Диалог выбора цвета

Диалоговое окно для выбора цвета из палитры.



Нажмите **Определить цвет**, чтобы расширить палитру.



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа [2.1.13.2. Результат выбора цвета \(стр. 55\)](#), в котором будет сохранено имя и тип выбранного файла. Ссылка на созданный экземпляр указывается в свойстве Ссылка на результат ([стр. 79](#)).

Пример использования приведен в [1.4.1. Выбор цвета на панели инструментов из палитры или из меню \(стр. 9\)](#).

2.2.5.1. Свойства

Палитра (PaletteType)

Вид палитры. Тип значения - уведомляющий uint1.

Может принимать только одно из допустимых значений, приведенных в таблице.

Допустимое значение	Имя значения	Результат выбора значения
«0»	Градиент	
«1»	Веб-цвета	

Расширенный диалог (IsExpanded)

Позволяет при вызове окна **ColorDialog** открывать сразу расширенный режим определения цвета. Тип значения - уведомляющий bool.

Значение:

- «true» - запускается расширенный режим определения цвета;
- «false» - запускается обычный режим определения цвета

Выбранный цвет (SelectedColor)

Позволяет выбрать цвет по умолчанию. При запуске окна **ColorDialog** в палитре будет выбран указанный цвет. Тип значения - uint4.

Ссылка на результат (Result)

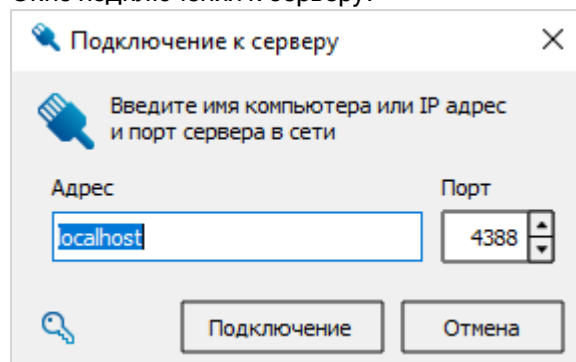
Ссылка на [2.1.13.2. Результат выбора цвета \(стр. 55\)](#), в котором будет сохранен результат выбора цвета.

OkAction

Ссылка на команду, выполняемую при нажатии на кнопку **OK**.

2.2.6. Диалог подключения к серверу

Окно подключения к серверу.



Чтобы указать учетные данные при подключении к защищенному паролем серверу, нажмите кнопку **Указать учетные данные**. Откроется окно [2.2.9.3. Диалог ввода учетных данных \(стр. 97\)](#).



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа [2.1.13.3. Результат выбора параметров подключения \(стр. 56\)](#), в котором будут сохранены данные подключения. Ссылка на созданный экземпляр указывается в свойстве `pResult` ([стр. 81](#)).

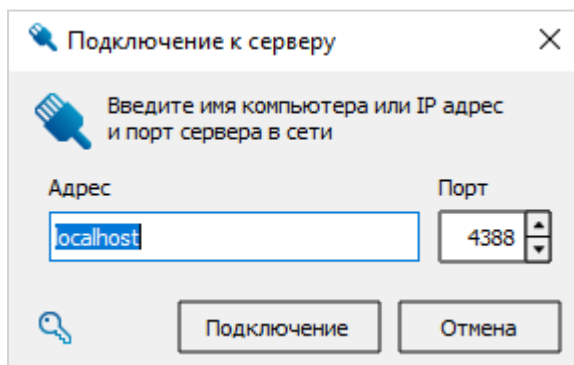
2.2.6.1. Свойства

Указать название (SpecifyTitle)

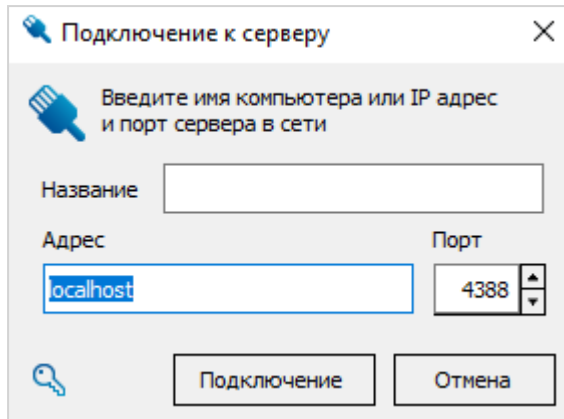
Позволяет указывать в окне подключения к серверу название подключения. Тип значения - bool.

Значение:

- «false» – обычный вид окна подключения к серверу;



- «true» - в окне появляется поле ввода названия подключения.

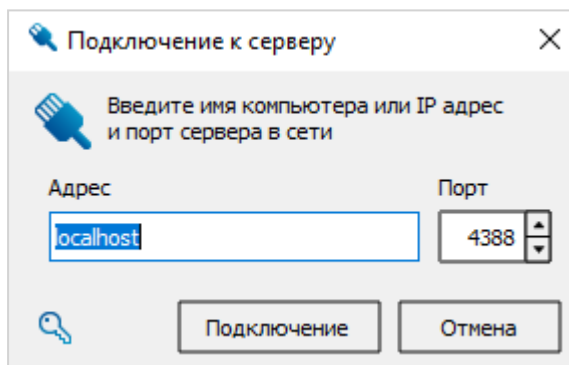


Указать порт истории (SpecifyHistoryPort)

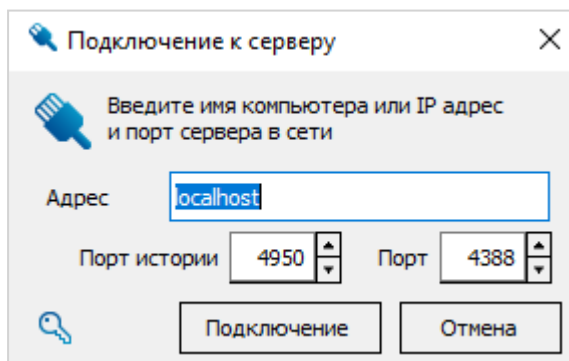
Позволяет указывать в окне подключения к серверу номер порта истории сервера. Тип значения - bool.

Значение:

- «false» - обычный вид окна подключения к серверу;



- «true» - в окне подключения появляется поле для ввода номера порта истории.



pResult

Ссылка на [2.1.13.3. Результат выбора параметров подключения \(стр. 56\)](#), в котором будут сохранены данные подключения.

OkAction

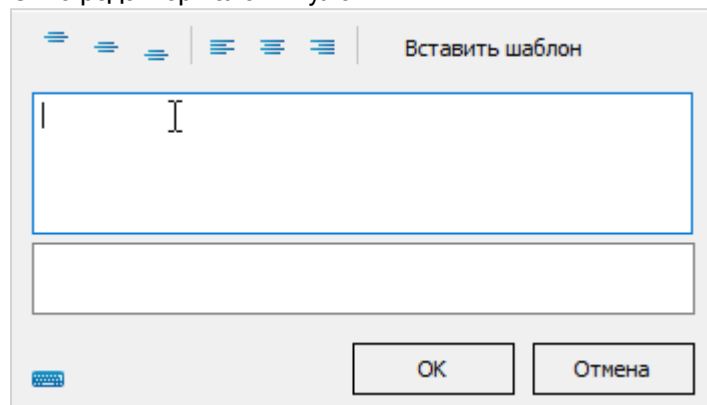
Ссылка на команду, выполняемую при нажатии на кнопку **Подключение** в окне подключения к серверу.

Вызов экранной клавиатуры (init_ScreenKeyboard)

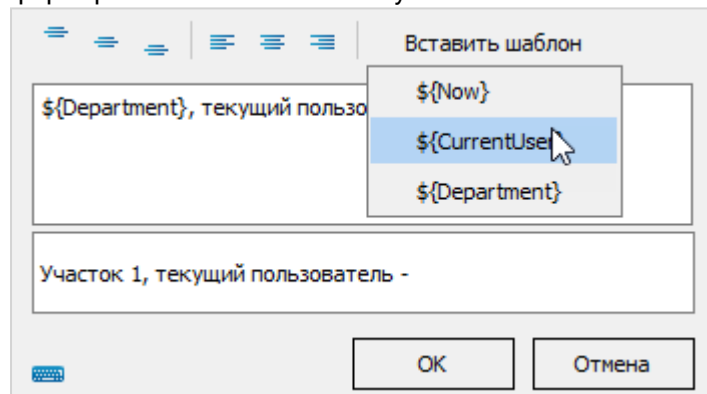
Ссылка на экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), хранящий настройки вызова экранной клавиатуры.

2.2.7. Редактор колонтитула

Окно-редактор колонтитулов.



Колонтитулы создают, например, для таблиц, при сохранении в файл или при печати на принтере. Для формирования текста колонтитулов можно использовать шаблоны.



Шаблоны создаются в экземпляре компонента [2.1.8. Текстовые шаблоны \(стр. 30\)](#). Чтобы указать набор шаблонов, который должен быть доступен для выбора в окне редактора, сделайте ссылку на экземпляр [2.1.8. Текстовые шаблоны \(стр. 30\)](#) из свойства Шаблоны (pTemplates) [\(стр. 83\)](#).



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа [2.1.13.6. Результат редактирования колонтитула \(стр. 59\)](#), в котором будет сохранен результат редактирования колонтитула. Ссылка на созданный экземпляр указывается в свойстве Ссылка на результат [\(стр. 83\)](#).

2.2.7.1. Свойства

Вызов экранной клавиатуры (init_ScreenKeyboard)

Ссылка на экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), хранящий настройки вызова экранной клавиатуры.

Шаблоны (pTemplates)

Ссылка на экземпляр типа [2.1.8. Текстовые шаблоны \(стр. 30\)](#), которые будут использоваться для подстановки в текст колонтитула.

pResult

Ссылка на экземпляр типа [2.1.13.6. Результат редактирования колонтитула \(стр. 59\)](#), в котором будет сохранен результат редактирования колонтитула.

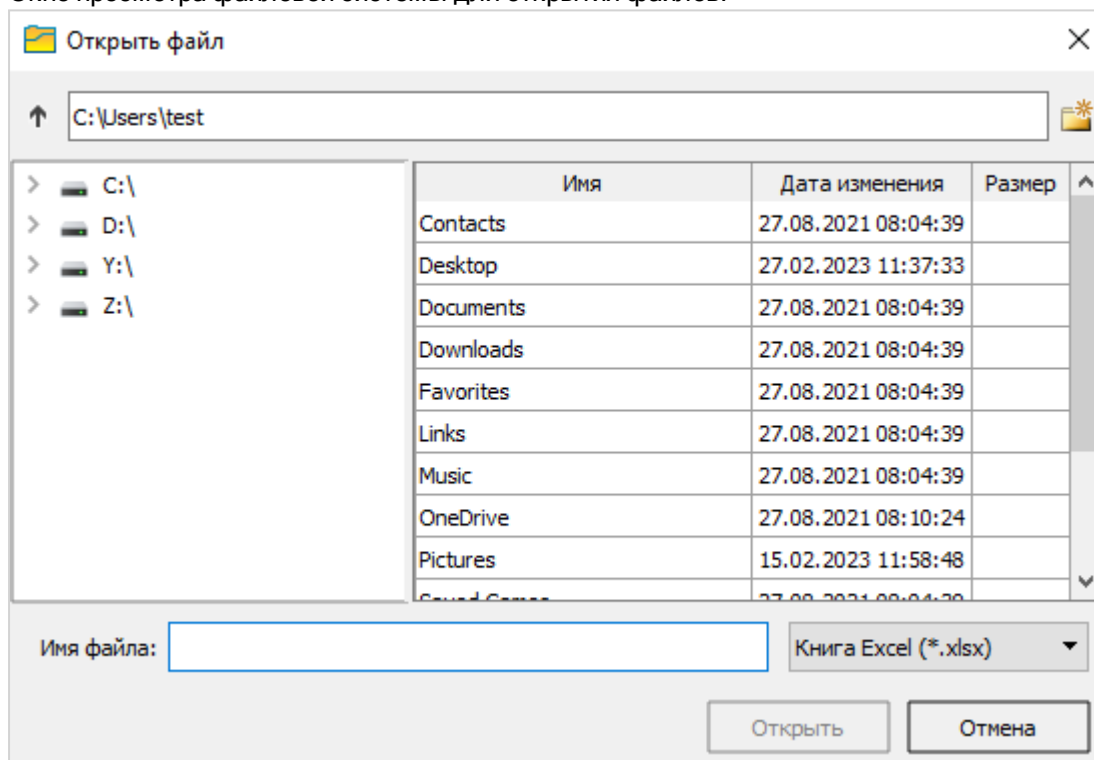
OkAction

Ссылка на команду, выполняемую при нажатии на кнопку **ОК**.

2.2.8. Файлы

2.2.8.1. Диалог открытия файла

Окно просмотра файловой системы для открытия файлов.



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа [2.1.13.7. Результат выбора файла \(стр. 59\)](#), в котором будет сохранено имя и тип выбранного файла. Ссылка на созданный экземпляр указывается в свойстве Ссылка на результат [\(стр. 87\)](#).

Свойства

OnPathChosen

Ссылка на команду, выполняемую в момент выбора строки файла в списке файлов окна [2.2.8.1. Диалог открытия файла \(стр. 83\)](#).

Корневой путь (RootPath)

Ограничивает область просмотра файловой системы. Тип значения - string.

Путь по умолчанию (DefaultPath)

Путь по умолчанию. Если указать значение, то при открытии окна [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) всегда будет открыт указанный каталог. Тип значения - string.

Имя файла по умолчанию (DefaultFileName)

Имя файла по умолчанию. Значение свойства подставляется в поле **Имя файла** при открытии окна [2.2.8.1. Диалог открытия файла \(стр. 83\)](#). Тип значения - string.


- Если в значении указать расширение файла (например «example.xml»), значение будет подставлено в поле в исходном виде.
- Если в значении не указывать расширение (например «example»), к значению будет добавлено расширение, выбранное из [FileTypeFilter \(Фильтр типа файлов\)](#).

Фильтр типа файлов (FileTypeFilter)

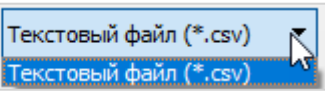
Фильтр отображаемого типа файлов. Тип значения - уведомляющий string.

Формат указания значения: «**Описание | расширение**». Чтобы указать несколько типов, перечислите их через вертикальную черту («|»).
Пример: «Текстовый файл (*.csv) | *.csv».

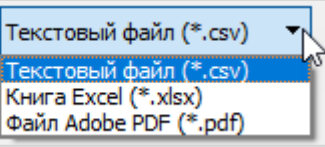
- Если указать значение, то в нижней части окна [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) будет доступен выпадающий список - фильтр отображения файлов.

 **ПРИМЕР**

«Текстовый файл (*.csv) | *.csv»

Тип файла: 

«Текстовый файл (*.csv) | *.csv | Книга Excel (*.xlsx) | *.xlsx | Файл Adobe PDF (*.pdf) | *.pdf»

Тип файла: 

- Если оставить значение свойства пустым, то при открытии окна будут отображаться все файлы и папки. Выпадающий список с фильтром будет недоступен.

Индекс типа в фильтре (DefaultFileType)

Устанавливает тип файлов по умолчанию в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#). Тип значения - int4.

Значение - номер записи типа в свойстве **FileTypeFilter (Фильтр типа файлов)** (нумерация ведется с «0»).

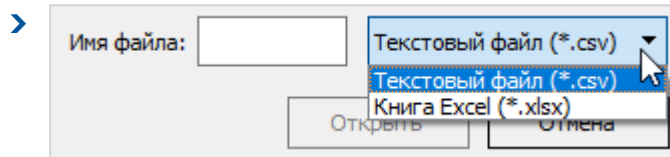


ПРИМЕР

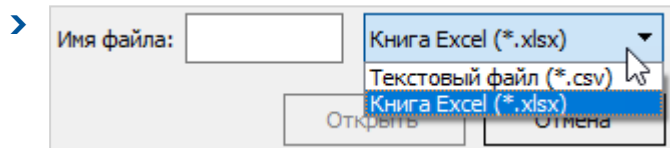
Допустим, в качестве значения свойства **FileTypeFilter (Фильтр типа файлов)** указана строка: «Текстовый файл (*.csv)|*.csv|Книга Excel (*.xlsx)|*.xlsx»

Тогда для **DefaultFileType** можно указать:

- «0» - при открытии окна [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) по умолчанию будет предложено открыть файл типа «Текстовый файл (*.csv)»



- «1» - при открытии окна [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) по умолчанию будет предложено открыть файл типа «Книга Excel (*.xlsx)|*.xlsx»



Показывать свойства файлов (ShowFileProperties)

Позволяет отображать больше информации о файлах в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#). Тип значения - уведомляющий bool.

Значение:

- «true» - отображаются имена, даты изменения и размеры файлов (в байтах);

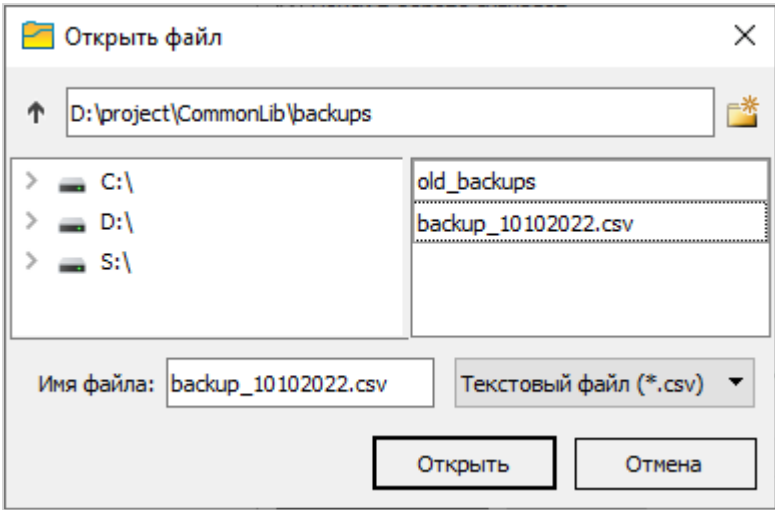
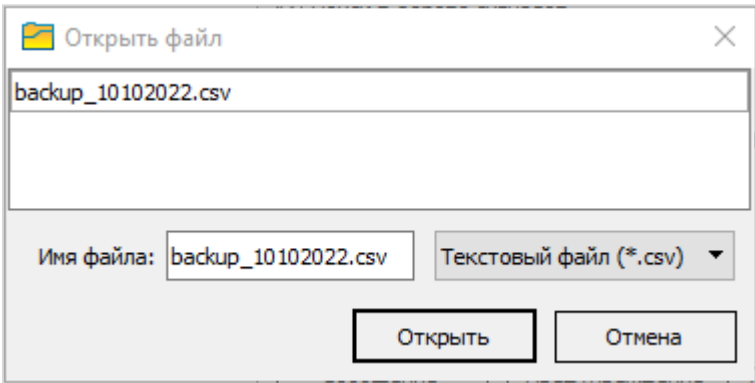
Имя	Дата изменения	Размер
old_backups	17.11.2022 14:12:29	
17112022_bckp.csv	17.11.2022 14:12:53	6377
18112022_bckp.csv	17.11.2022 14:12:53	6377

- «false» - отображаются только имена файлов.

old_backups
17112022_bckp.csv
18112022_bckp.csv

Разрешена смена папки (ChangeFolderAllowed)

Указывает на возможность смены папки, открываемой в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) по умолчанию. Тип значения - уведомляющий bool.

Значение	Результат
«true» - разрешено менять папку	<p>В этом случае в окне 2.2.8.1. Диалог открытия файла (стр. 83) отображается дерево каталогов и их содержимое (файлы и другие папки). Есть возможность перемещаться между каталогами и создавать новые папки.</p> 
«false» - запрещено менять папку	<p>В этом случае в окне 2.2.8.1. Диалог открытия файла (стр. 83) отображаются только файлы папки, открываемой по умолчанию. Дерево каталогов скрыто, нет возможности создавать новые папки.</p> 

Разрешено создание папок (CreateFolderAllowed)

Указывает на возможность создания папок в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#). Тип значения - уведомляющий bool.

Значение:

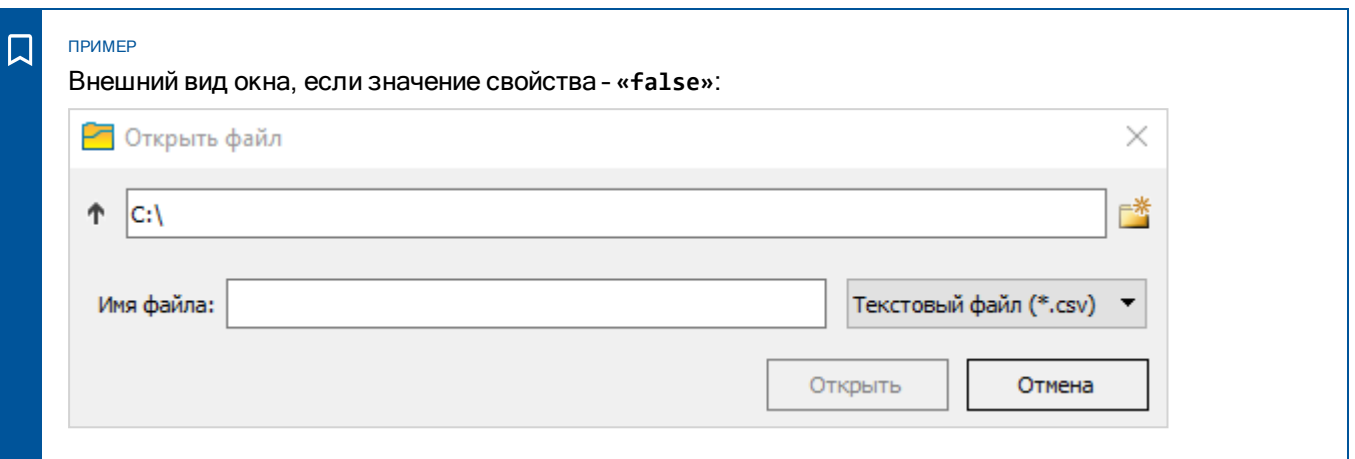
- «true» - разрешено создавать папки;
- «false» - запрещено создавать папки.

Отображать панель просмотра (ShowBrowsePanel)

Ограничивает возможность просмотра иерархии папок и уменьшает размеры диалога. Тип значения - уведомляющий bool.

Значение:

- «true» - просмотр иерархии папок разрешен;
- «false» - просмотр иерархии папок запрещен.



Ссылка на результат (Result)

Ссылка на экземпляр типа [2.1.13.7. Результат выбора файла \(стр. 59\)](#), в котором будет сохранено имя и тип выбранного файла.

Отображать кнопку Параметры (EditSettingsAllowed)

Меняет видимость кнопки **Параметры** в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#). Тип значения - уведомляющий bool.

Значение:

- «true» - кнопка есть;
- «false» - кнопки нет.

При нажатии на кнопку активируется команда OnEditSettings.

OnEditSettings

Ссылка на команду, выполняемую при нажатии на кнопку **Параметры** в окне [2.2.8.1. Диалог открытия файла \(стр. 83\)](#).

Видимость кнопки зависит от значения свойства [EditSettingsAllowed \(Отображать кнопку Параметры\)](#).

OnComplete

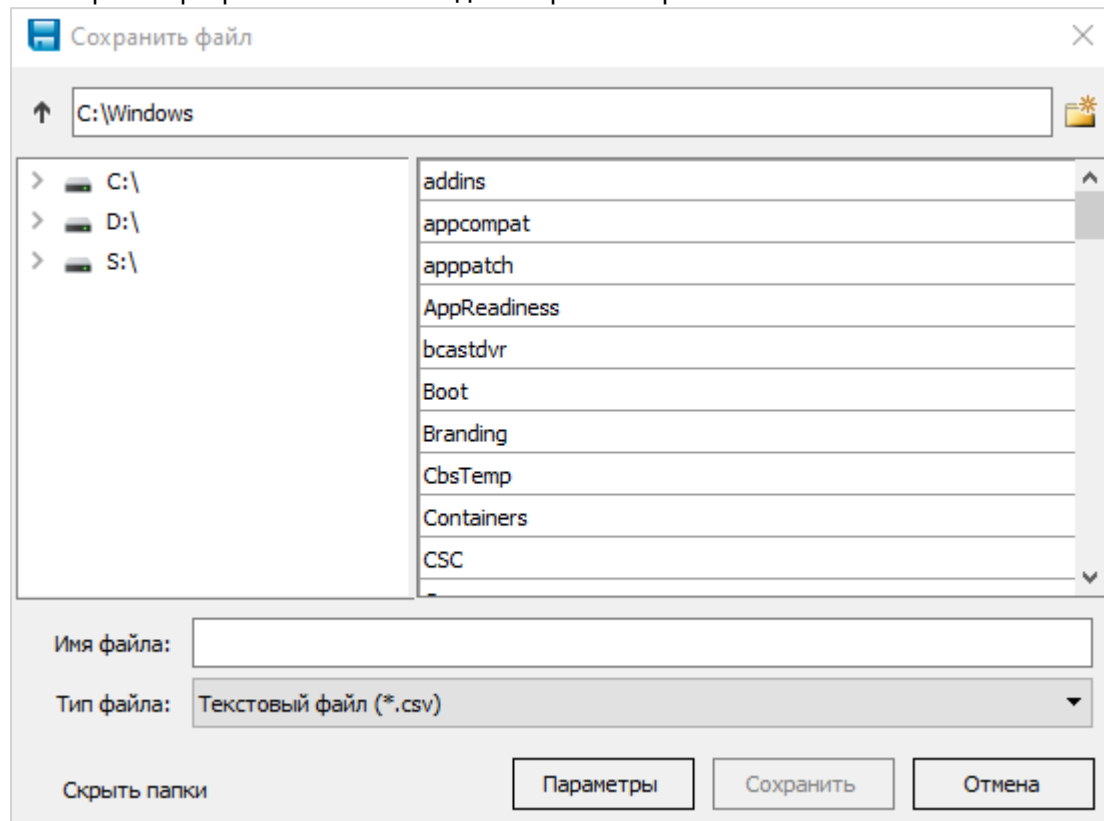
Ссылка на команду, выполняемую в результате нажатия кнопки **Открыть** и закрытии окна [2.2.8.1. Диалог открытия файла \(стр. 83\)](#).

Вызов экранной клавиатуры (init_ScreenKeyboard)

Ссылка на экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), хранящий настройки вызова экранной клавиатуры.

2.2.8.2. Диалог сохранения файла

Окно просмотра файловой системы для сохранения файлов.



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа [2.1.13.7. Результат выбора файла \(стр. 59\)](#), в котором будет сохранено имя и тип выбранного файла. Ссылка на созданный экземпляр указывается в свойстве Ссылка на результат ([стр. 92](#)).

Свойства

OnPathChosen

Ссылка на команду, выполняемую в момент выбора строки файла в списке файлов окна [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#).

Корневой путь (RootPath)

Ограничивает область просмотра файловой системы. Тип значения - string.

Путь по умолчанию (DefaultPath)

Путь по умолчанию. Если указать значение, то при открытии окна [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#) всегда будет открыт указанный каталог. Тип значения - string.

Имя файла по умолчанию (DefaultFileName)

Имя файла по умолчанию. Значение свойства подставляется в поле **Имя файла** при открытии окна [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#). Тип значения - string.

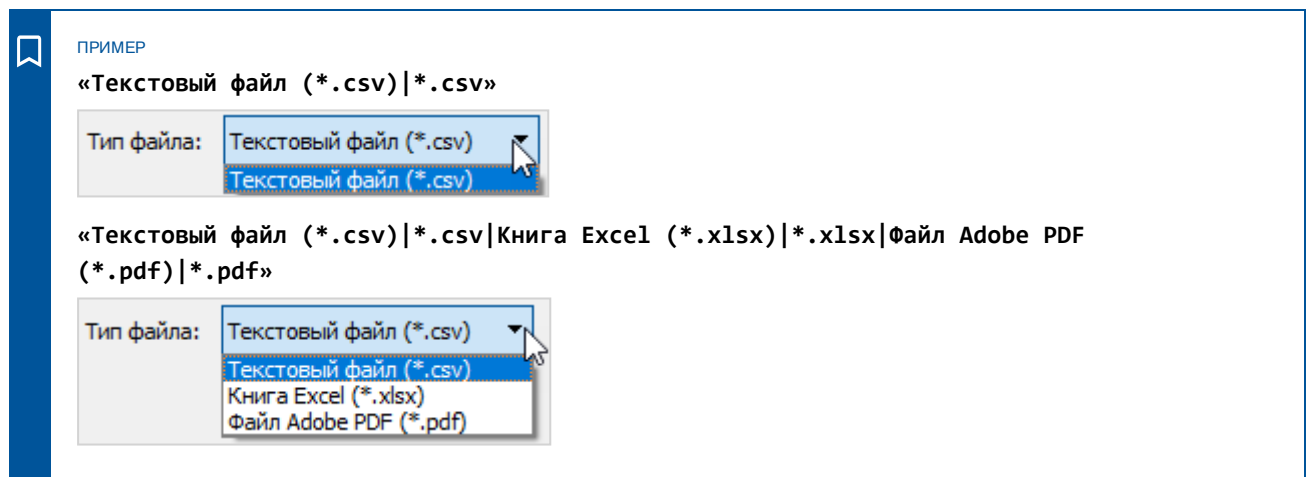
- Если в значении указать расширение файла (например «example.xml»), значение будет подставлено в поле в исходном виде.
- Если в значении не указывать расширение (например «example»), к значению будет добавлено расширение, выбранное из **FileTypeFilter (Фильтр типа файлов)**.

Фильтр типа файлов (FileTypeFilter)

Фильтр отображаемого типа файлов. Тип значения - уведомляющий string.

Формат указания значения: «Описание | расширение». Чтобы указать несколько типов, перечислите их через вертикальную черту («|»).

- Если указать значение, то в нижней части окна [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#) будет доступен выпадающий список - фильтр отображения файлов.



- Если оставить значение свойства пустым, то при открытии окна будут отображаться все файлы и папки. Выпадающий список с фильтром будет недоступен.

Индекс типа в фильтре (DefaultFileType)

Устанавливает тип файлов по умолчанию в окне [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#). Тип значения - int4.

Значение - номер записи типа в свойстве **FileTypeFilter (Фильтр типа файлов)** (нумерация ведется с «0»).

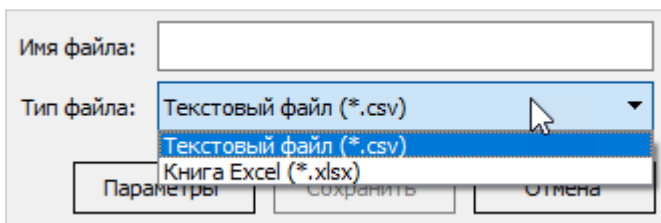


ПРИМЕР

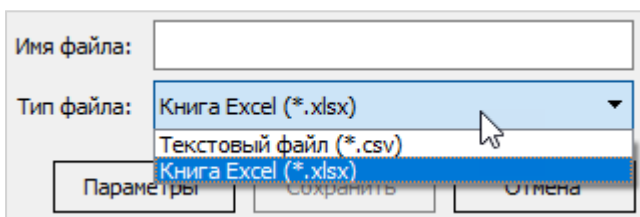
Допустим, в качестве значения свойства **FileTypeFilter (Фильтр типа файлов)** указана строка:
«Текстовый файл (*.csv)|*.csv|Книга Excel (*.xlsx)|*.xlsx»

Тогда для **DefaultFileType** можно указать:

- «0» - при открытии окна [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#) по умолчанию будет предложено сохранить файл типа «Текстовый файл (*.csv)»



- «1» - при открытии окна [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#) по умолчанию будет предложено сохранить файл типа «Книга Excel (*.xlsx)|*.xlsx»



Показывать свойства файлов (ShowFileProperties)

Позволяет отображать больше информации о файлах в окне [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#). Тип значения - уведомляющий bool.

Значение:

- «true» - отображаются имена, даты изменения и размеры файлов (в байтах);

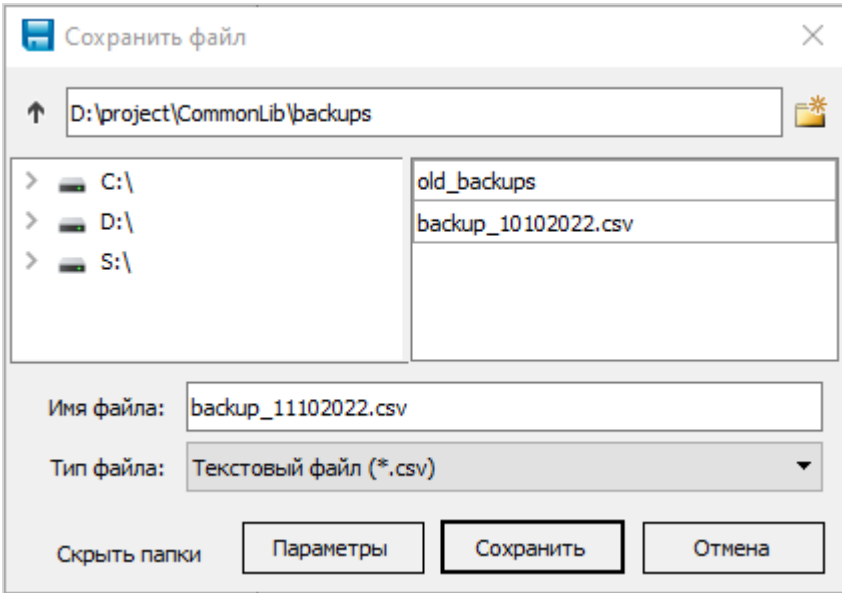
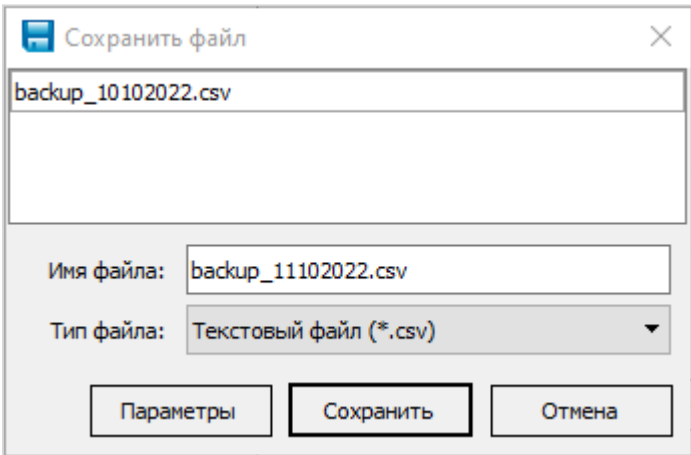
Имя	Дата изменения	Размер
old_backups	17.11.2022 14:12:29	
17112022_bckp.csv	17.11.2022 14:12:53	6377
18112022_bckp.csv	17.11.2022 14:12:53	6377

- «false» - отображаются только имена файлов.

old_backups
17112022_bckp.csv
18112022_bckp.csv

Разрешена смена папки (ChangeFolderAllowed)

Указывает на возможность смены папки, открывающейся в окне [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#) по умолчанию. Тип значения - уведомляющий bool.

Значение	Результат
«true» - разрешено менять папку	<p>В этом случае в окне 2.2.8.2. Диалог сохранения файла (стр. 88) отображается дерево каталогов и их содержимое (файлы и другие папки). Есть возможность перемещаться между каталогами и создавать новые папки.</p> 
«false» - запрещено менять папку	<p>В этом случае в окне 2.2.8.2. Диалог сохранения файла (стр. 88) отображаются только файлы папки, открываемой по умолчанию. Дерево каталогов скрыто, нет возможности создавать новые папки.</p> 

Разрешено создание папок (CreateFolderAllowed)

Указывает на возможность создания папок в окне [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#). Тип значения - уведомляющий bool.

Значение:

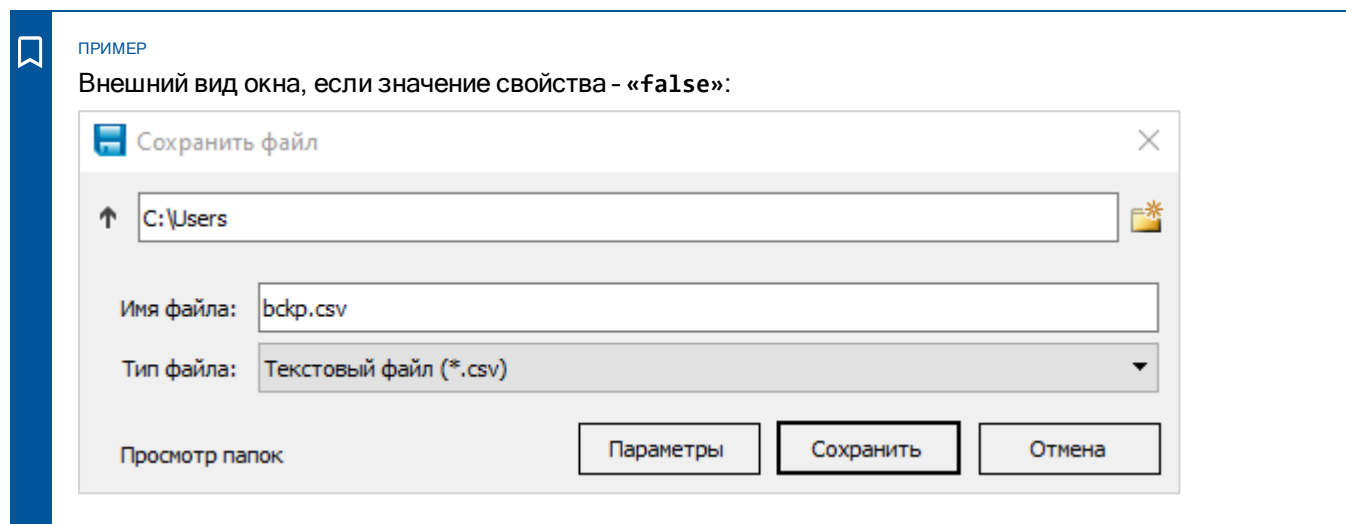
- «true» - разрешено создавать папки;
- «false» - запрещено создавать папки.

Отображать панель просмотра (ShowBrowsePanel)

Ограничивает возможность просмотра иерархии папок и уменьшает размеры диалога. Тип значения - уведомляющий bool.

Значение:

- «true» - просмотр иерархии папок разрешен;
- «false» - просмотр иерархии папок запрещен.



Ссылка на результат (Result)

Ссылка на экземпляр типа [2.1.13.7. Результат выбора файла \(стр. 59\)](#), в котором будет сохранено имя и тип сохраняемого файла.

Отображать кнопку Параметры (EditSettingsAllowed)

Меняет видимость кнопки **Параметры** в окне [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#). Тип значения - уведомляющий bool.

Значение:

- «true» - кнопка есть;
- «false» - кнопки нет.

При нажатии на кнопку активируется команда OnEditSettings.

OnEditSettings

Ссылка на команду, выполняемую при нажатии на кнопку **Параметры** в окне [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#).

Видимость кнопки зависит от значения свойства [EditSettingsAllowed \(Отображать кнопку Параметры\)](#).

OnComplete

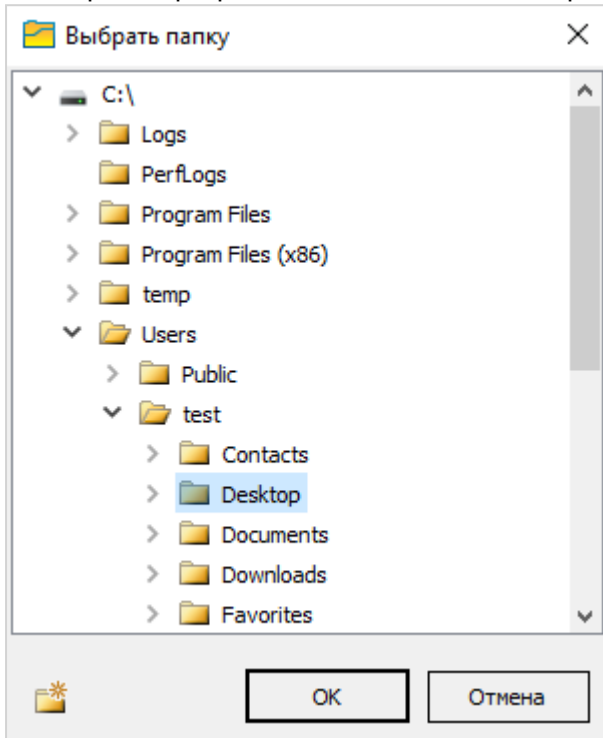
Ссылка на команду, выполняемую в результате нажатия кнопки **Сохранить** и закрытии окна [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#).

Вызов экранной клавиатуры (init_ScreenKeyboard)

Ссылка на экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), хранящий настройки вызова экранной клавиатуры.

2.2.8.3. Диалог выбора папки

Окно просмотра файловой системы для выбора папки.



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа [2.1.13.1. Результат выбора папки \(стр. 55\)](#), в котором будет сохранен результат выбора папки. Ссылка на созданный экземпляр указывается в свойстве Result [\(стр. 94\)](#).

Свойства

Корневой путь (RootPath)

Ограничивает область просмотра файловой системы. Тип значения - string.

Путь по умолчанию (DefaultPath)

Путь по умолчанию. Если указать значение, то при открытии окна [2.2.8.3. Диалог выбора папки \(стр. 93\)](#) всегда будет открыт указанный каталог. Тип значения - string.

Разрешено создание папок (CreateFolderAllowed)

Указывает возможность создания папок в окне [2.2.8.3. Диалог выбора папки \(стр. 93\)](#). Тип значения - уведомляющий bool.

Значение:

- «true» - разрешено создавать папки;
- «false» - запрещено создавать папки.

Result

Ссылка на экземпляр типа [2.1.13.1. Результат выбора папки \(стр. 55\)](#), в котором будет сохранен путь к выбранной папке.

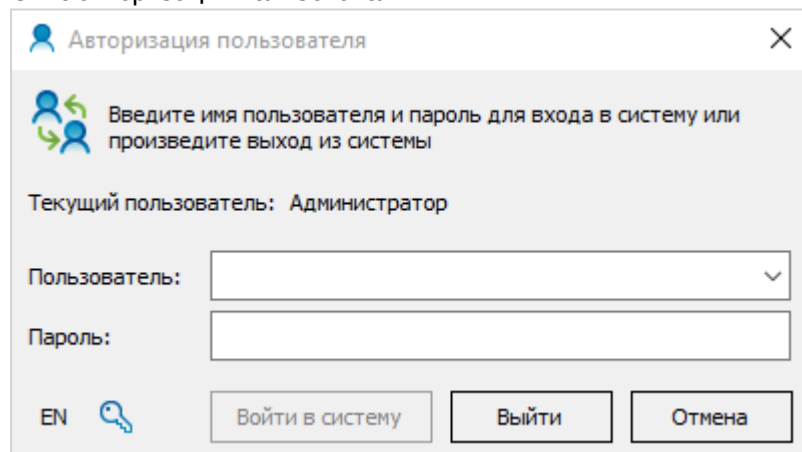
OnComplete

Ссылка на команду, вызываемую при закрытии окна [2.2.8.3. Диалог выбора папки \(стр. 93\)](#) после выбора папки.

2.2.9. Безопасность

2.2.9.1. Диалог авторизации пользователя

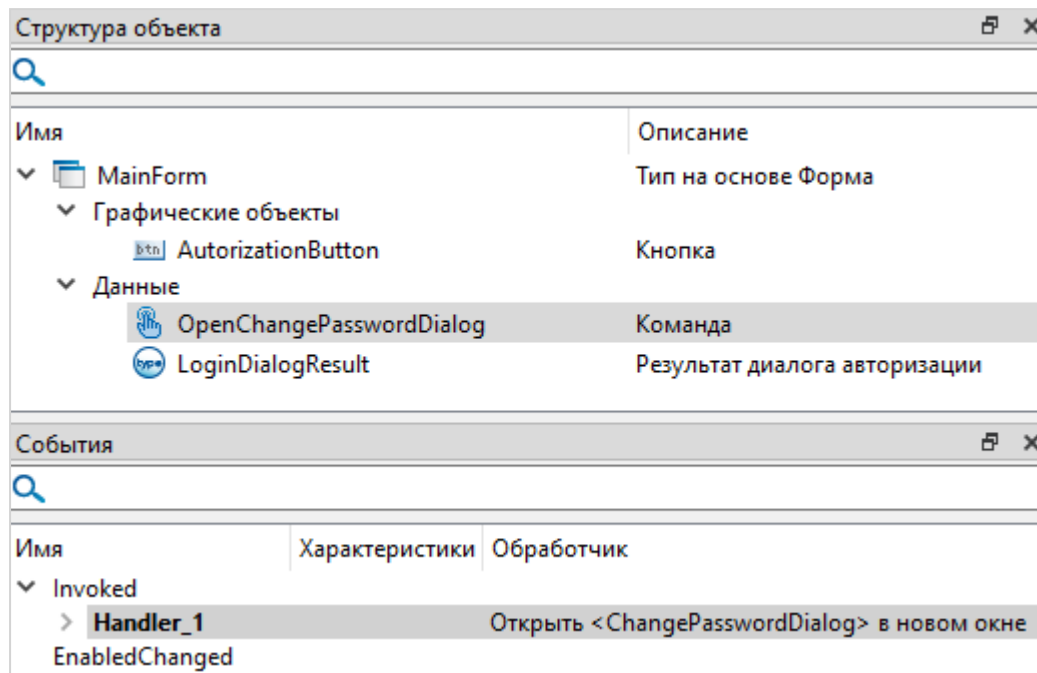
Окно авторизации пользователя.



По умолчанию кнопка смены пароля в нижнем левом углу окна не работает. Чтобы по нажатию кнопки открывалось окно смены пароля [2.2.9.2. Диалог смены пароля \(стр. 96\)](#), выполните следующие действия:

1. Создайте кнопку, по нажатию которой должно открываться окно авторизации `LoginDialog`. Настройте открытие окна в обработчике события `ButtonPressed` (или `MouseClicked` - для других визуальных объектов).
2. Создайте экземпляр типа [2.1.13.8. Результат диалога авторизации \(стр. 60\)](#). Он необходим для передачи данных о текущем пользователе между окнами.

3. Создайте экземпляр команды (располагается в стандартной библиотеке компонентов в юните **Встроенные**). В обработчике события **Invoked** созданной команды настройте открытие окна [2.2.9.2. Диалог смены пароля \(стр. 96\)](#).



4. Затем разверните настроенный обработчик события. Найдите в открывшемся списке свойства **UserId** и **UserDisplayName**. В качестве значений этих свойств укажите ссылки на соответствующие свойства экземпляра [2.1.13.8. Результат диалога авторизации \(стр. 60\)](#), созданного на втором шаге.

5. Вернитесь к кнопке открытия окна авторизации **LoginDialog**. Разверните обработчик открытия окна и найдите в открывшемся списке свойств ссылку на команду **ChangePassword**. Сошлитесь здесь на команду, созданную на третьем шаге. Здесь же в качестве значения свойства **Ссылка на результат** укажите ссылку на экземпляр [2.1.13.8. Результат диалога авторизации \(стр. 60\)](#), созданный на втором шаге.

Чтобы не выполнять описанную выше настройку, используйте кнопку вызова окна авторизации [2.1.12.1. Кнопка авторизации \(стр. 51\)](#). Для полноценной работы этого компонента не нужны дополнительные настройки.

Свойства

Ссылка на результат (Result)

Ссылка на [2.1.13.8. Результат диалога авторизации \(стр. 60\)](#), в котором будут сохранены id и отображаемое имя текущего пользователя.

ChangePassword

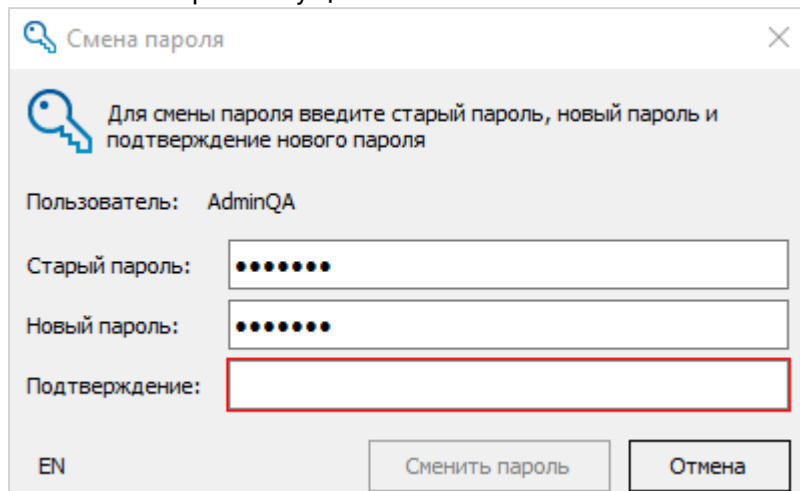
Ссылка на команду, открывающую окно [2.2.9.2. Диалог смены пароля \(стр. 96\)](#).

Вызов экранной клавиатуры (init_ScreenKeyboard)

Ссылка на экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), хранящий настройки вызова экранной клавиатуры.

2.2.9.2. Диалог смены пароля

Окно смены пароля текущего пользователя.



Пример применения приведен в описании компонента [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#). Также может использоваться самостоятельно.

Свойства

UserId

Хранит уникальный идентификатор (uid) текущего пользователя подсистемы безопасности SePlatform.Security. Тип значения - string.



ПРИМЕР

Вид значения: alpha:xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx.

Пример значения: «a1pha:c3a997b1-326e-42ee-8907-e359fc17feb7».

UserDisplayName

Хранит отображаемое имя текущего пользователя подсистемы безопасности SePlatform.Security. Тип значения - string.



ПРИМЕР

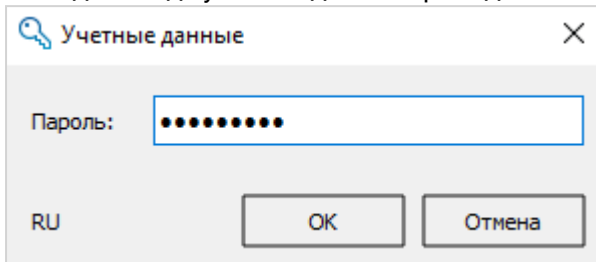
Пример значения: «Иванов».

Вызов экранной клавиатуры (init_ScreenKeyboard)

Ссылка на экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), хранящий настройки вызова экранной клавиатуры.

2.2.9.3. Диалог ввода учетных данных

Окно для ввода учетных данных при подключении к защищенному паролем серверу.



Используется в окне подключения к серверу [2.2.6. Диалог подключения к серверу \(стр. 80\)](#), однако также может использоваться самостоятельно.



ПРИМЕЧАНИЕ

Используя компонент, создайте экземпляр типа **Учетные данные AP**, в котором будут сохранены данные подключения. Ссылка на созданный экземпляр указывается в свойстве `pCredentials` ([стр. 97](#)).

Свойства

pCredentials

Ссылка на экземпляр типа **Учетные данные AP**, в который будут помещены введенные учетные данные.

OkAction

Ссылка на команду, выполняемую при нажатии на кнопку **ОК**.

Вызов экранной клавиатуры (init_ScreenKeyboard)

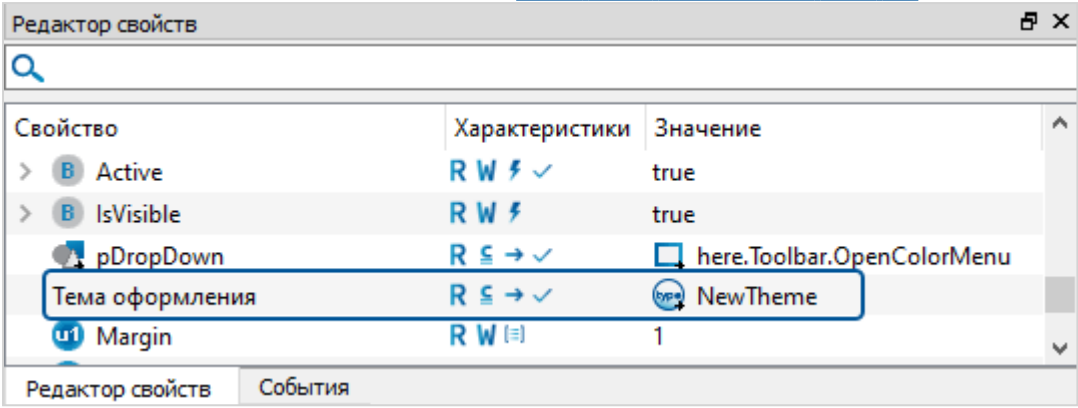
Ссылка на экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), хранящий настройки вызова экранной клавиатуры.

2.3. Общие свойства компонентов библиотеки

2.3.1. Тема оформления (pTheme)

Тема оформления визуальных компонентов SePlatform.HMI.CommonLib.

Тип значения - ссылка на экземпляр типа [2.1.1. Тема оформления \(стр. 12\)](#).



История изменений

2.0

Изменения

- Англоязычные названия компонентов заменены русскоязычными.
- Для корректной работы компонента [2.1.14. Дерево сигналов \(стр. 63\)](#) (ItemTree) в свойствах **pDragStarted** и **pDoubleClicked** вместо ссылок на экземпляр типа **Команда** необходимо указывать ссылки на экземпляр нового типа [2.1.7. Обработчик события \(стр. 28\)](#), описанный в блоке "Новые возможности".
- У компонентов [2.1.9.3. Редактор времени \(стр. 34\)](#) (TimeEditor), [2.1.9.2. Редактор даты \(стр. 33\)](#) (DateEditor) и [2.1.9.4. Редактор интервала времени \(стр. 35\)](#) (IntervalEditor) удалено свойство **IsVisible**. Вместо него достаточно использовать стандартное свойство **Visible**.
- Компонент [2.1.9.4. Редактор интервала времени \(стр. 35\)](#) (IntervalEditor) теперь использует шрифт и цвет фона из указанного экземпляра типа **Тема**.

Новые возможности

- Созданы новые компоненты библиотеки:
 - [2.1.7. Обработчик события \(стр. 28\)](#), позволяющий создать пользовательское событие с возможностью передачи его аргументов;
 - [2.2.7. Редактор колонтитула \(стр. 82\)](#), позволяющий создавать колонтитулы, например, для таблиц, при сохранении в файл или при печати на принтере;
 - Тип [2.1.8. Текстовые шаблоны \(стр. 30\)](#), представляющий собой набор текстовых шаблонов, которые могут быть подставлены, например, в текст колонтитула;
 - Компоненты [2.1.9.5. Горизонтальный сплиттер \(стр. 36\)](#) и [2.1.9.6. Вертикальный сплиттер \(стр. 39\)](#), предназначенные для разделения области пользовательского интерфейса на две части с возможностью изменять размеры этих частей с помощью мышки.
- Расширены возможности применения компонента [2.1.1. Тема оформления \(стр. 12\)](#):
 - Добавлены свойства для цветового оформления панели инструментов;
 - Добавлены свойства для цветового оформления меню.
- Добавлен вспомогательный тип [2.1.11.5. Тема оформления панели инструментов \(стр. 49\)](#), позволяющий менять тему оформления визуальных компонентов, добавленных на панель инструментов.

Исправления

- Добавлена проверка существования папки в диалогах открытия и сохранения файлов. Ранее при указании пустого пути или несуществующей папки была доступна кнопка **Сохранить**, а при ее нажатии в разных случаях появлялись сообщения об успешном сохранении файла или об ошибке.
- Устранена причина, по которой экранная клавиатура в AstraLinux открывалась только при первом нажатии [2.1.12.5. Кнопка вызова экранной клавиатуры \(стр. 53\)](#).

2.0.1

Новые возможности

- Для компонента [2.1.14. Дерево сигналов \(стр. 63\)](#) реализованы функции **Find(pattern: string)** и **CancelFind()**. Первая предназначена для поиска сигналов в дереве по шаблону названия, вторая - для

отмены поиска.

- Компоненту [2.1.9.4. Редактор интервала времени \(стр. 35\)](#) добавлено свойство-ссылка на событие **pShiftCompleted**. Указанное событие активируется только при смещении интервала времени с помощью кнопок "Перейти к предыдущему интервалу времени" и "Перейти к следующему интервалу времени".

Улучшения

Размеры некоторых компонентов увеличены для удобства использования на сенсорной панели. Среди них:

- ширина [2.1.11.2. Кнопка выпадающего списка \(стр. 46\)](#);
- ширина кнопок изменения значения в [2.1.9.1. Редактор целого числа \(стр. 32\)](#), [2.1.9.3. Редактор времени \(стр. 34\)](#) и [2.1.9.4. Редактор интервала времени \(стр. 35\)](#);
- ширина кнопки вызова календаря в [2.1.9.2. Редактор даты \(стр. 33\)](#);
- толщина [2.1.9.5. Горизонтальный сплиттер \(стр. 36\)](#) и [2.1.9.6. Вертикальный сплиттер \(стр. 39\)](#).

Исправления

- [2.1.14. Дерево сигналов \(стр. 63\)](#) теперь очищается в процессе подключения к новому серверу.
- Пустое [2.2.1. Меню \(стр. 70\)](#) теперь не открывается. Ранее меню, не содержащее элементов, открывалось в виде короткой полосы толщиной в один пиксель.

2.0.2

Новая возможность

Разработана новая функция **GetMonitorForRect()** компонента [2.1.4. Утилиты для работы с окном \(стр. 22\)](#), служащая для определения монитора, на котором отображается большая часть прямоугольника. Может использоваться для открытия дочерней формы в центре родительского компонента.

Улучшение

Улучшено выполнение функции **FitToMonitor()** компонента [2.1.4. Утилиты для работы с окном \(стр. 22\)](#): при размещении окна на указанном мониторе теперь учитываются примерные толщина границ и высота заголовка окна.

Исправление

Поиск в дереве сигналов по названию папки выполнялся некорректно: при раскрытии найденной папки её содержимое помещалось в корень дерева.

Изменения документации

Редакция 1

В руководстве пользователя обновлена иллюстрация юнита внешнего модуля **CommonLib**.

В справочном руководстве:

- Актуализированы названия компонентов библиотеки.
- Описаны новые компоненты [2.1.7. Обработчик события \(стр. 28\)](#), [2.2.7. Редактор колонтитула \(стр. 82\)](#), [2.1.8. Текстовые шаблоны \(стр. 30\)](#), [2.1.9.5. Горизонтальный сплиттер \(стр. 36\)](#) и [2.1.9.6. Вертикальный сплиттер \(стр. 39\)](#), [2.1.11.5. Тема оформления панели инструментов \(стр. 49\)](#).

- Обновлены описания компонентов [2.1.1. Тема оформления \(стр. 12\)](#), [2.1.9.3. Редактор времени \(стр. 34\)](#), [2.1.9.2. Редактор даты \(стр. 33\)](#) и [2.1.9.4. Редактор интервала времени \(стр. 35\)](#).
- Актуализирован пример возвращаемого значения функции `GetViewSettings()` компонента [2.1.1. Тема оформления \(стр. 12\)](#).

Редакция 2

В справочном руководстве:

- Описаны новые:
 - функции `Find(pattern: string)` и `CancelFind()` компонента [2.1.14. Дерево сигналов \(стр. 63\)](#).
 - свойство `pShiftCompleted` компонента [2.1.9.4. Редактор интервала времени \(стр. 35\)](#).
- Актуализировано описание компонента [2.1.11.5. Тема оформления панели инструментов \(стр. 49\)](#) и его свойств.
- Добавлены описания ранее не описанных свойств:
 - `pOnDoClose` компонента [2.2.1. Меню \(стр. 70\)](#);
 - **Учетные данные (Credentials)** компонента [2.1.13.3. Результат выбора параметров подключения \(стр. 56\)](#);
 - **Вызов экранной клавиатуры (init_ScreenKeyboard)** компонента [2.2.7. Редактор колонтитула \(стр. 82\)](#).
- Во всех описаниях свойств компонентов актуализированы названия и типы значений.
- Актуализированы скриншоты.

Редакция 3

- В справочном руководстве для компонента [2.1.4. Утилиты для работы с окном \(стр. 22\)](#) добавлены описания новой функции `GetMonitorForRect()` и ранее не описанной функции `FitFormSize()`.
- Обновлен проект-пример, прикладываемый к разделу [1.4.1. Выбор цвета на панели инструментов из палитры или из меню \(стр. 9\)](#) руководства пользователя. В проекте устранена ошибка, возникавшая в журнале времени исполнения, но не влиявшая на работоспособность проекта.
- Бывший раздел "Приложения" переименован в "Примеры использования компонентов расширения".

1.4

Новая возможность

Создан новый компонент - кнопка вызова экранной клавиатуры [2.1.12.5. Кнопка вызова экранной клавиатуры \(стр. 53\)](#). Экземпляр кнопки следует использовать совместно с новым типом - [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#). Кнопка добавлена в:

- диалоги авторизации (`LoginDialog`, `ChangePasswordDialog`);
- диалог подключения к серверу (`ConnectDialog`, `CredentialsDialog`);
- диалог ввода значения (`InputDialog`);
- диалоги открытия и сохранения файла (`OpenFileDialog`, `SaveFileDialog`).

Чтобы кнопка отобразилась в указанных диалоговых окнах, сошлитесь на экземпляр типа [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#), где в свойстве `Enabled (Включено)` укажите значение «true».

Улучшение

Всплывающая подсказка с текстом ошибки теперь отображается всегда, без необходимости наведения мыши на элемент с ошибкой, в следующих компонентах:

- в диалоге ввода значения (**InputDialog**);
- в диалоге подключения к серверу (**ConnectDialog**);
- в диалоге смены пароля (**ChangePasswordDialog**);
- в редакторе интервала времени (**IntervalEditor**);
- у индикатора подключения (**ConnectionIndicator**).

При этом для индикации ошибок ввода используется новый компонент - **ErrorFrame**. Он используется для отображения всплывающей подсказки под полем ввода без наведения мыши.

1.4.1

Исправления

- Календарь в редакторе даты [2.1.9.2. Редактор даты \(стр. 33\)](#) не окрашивался в цвета выбранной темы оформления.
- Всплывающая подсказка с текстом ошибки не всегда отображалась при изменении текста ошибки.
- Исключена вероятность сохранения файла с пустым именем в окне [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#), когда в поле ввода имени указана только часть пути к файлу.
- Устранена причина повторного входа в подсистему безопасности при использовании клавиши **Enter** в окне авторизации пользователя [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#).

1.4.2

Внутренние изменения. Функциональность компонента не изменилась.

1.4.3

Улучшение

Теперь при авторизации пользователя после изменения парольных политик в окне [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#) предлагается сменить пароль, если старый пароль не подходит новым парольным политикам. Ранее в таком случае возникала ошибка.

Изменения документации

Редакция 1

В руководстве пользователя обновлено [1.4.1. Выбор цвета на панели инструментов из палитры или из меню \(стр. 9\)](#) и проект-пример, прикладываемый к приложению.

В справочном руководстве:

- Описаны новые компоненты [2.1.12.5. Кнопка вызова экранной клавиатуры \(стр. 53\)](#) и [2.1.12.6. Параметры вызова экранной клавиатуры \(стр. 54\)](#).

- Для компонентов, предполагающих ввод значений, описано новое свойство [init_ScreenKeyboard \(Вызов экранной клавиатуры\)](#), позволяющее отобразить кнопку вызова экранной клавиатуры. Это:
 - диалоги авторизации ([LoginDialog](#), [ChangePasswordDialog](#));
 - диалог подключения к серверу ([ConnectDialog](#), [CredentialsDialog](#));
 - диалог ввода значения ([InputDialog](#));
 - диалоги открытия и сохранения файла ([OpenFileDialog](#), [SaveFileDialog](#)).
- Обновлено описание компонента [2.1.13.2. Результат выбора цвета \(стр. 55\)](#). Ранее писалось, что компонент сам сохраняет пользовательскую палитру в файл `colors.json` в папке проекта. Это не так, пользовательская палитра сохраняется только в значении свойства [CustomColors](#). Как сохранить пользовательскую палитру в файл теперь описано в [1.4.1. Выбор цвета на панели инструментов из палитры или из меню \(стр. 9\)](#).
- Порядок разделов с описаниями компонентов изменился. Ранее разделы были расположены в порядке алфавита, теперь порядок соответствует интерфейсу библиотеки, подключенной к проекту SePlatform.HMI.

Редакция 2

В справочном руководстве:

- Обновлено описание компонентов [2.2.9.1. Диалог авторизации пользователя \(стр. 94\)](#), [2.2.9.2. Диалог смены пароля \(стр. 96\)](#), [2.1.13.8. Результат диалога авторизации \(стр. 60\)](#), [2.1.5. Информация о продукте \(стр. 24\)](#).
- Устаревшее название типа "ApSource" заменено на "Источник AP".
- Обновлено скриншоты.

1.3

Важно. Текущая версия предназначена для использования с SePlatform.HMI 2.0.

Новые возможности

- Разработан новый компонент - диалог указания учетных данных [2.2.9.3. Диалог ввода учетных данных \(стр. 97\)](#). Диалог используется для указания учетных данных при подключении к защищенному паролем серверу.
- Теперь элемент меню можно настроить так, чтобы в его правой части отображалось текстовое описание сочетания "горячих" клавиш (например, "Сохранить Ctrl+S").
- В диалоге открытия файлов [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) появилась кнопка **Параметры**. Видимость кнопки можно изменить в значении свойства [Отображать кнопку Параметры \(EditSettingsAllowed\)](#). Команда, выполняемая в результате нажатия кнопки, указывается в свойстве [OnEditSettings](#).

Исправления

- Дата в календаре, размещенном на форме, теперь отображается корректно.
- Устранено предупреждение об устаревшем элементе `DataSource.Browser.GetChildren`. Вместо метода [GetChildren](#) теперь используется [GetChildrenWithRequestId](#), который стал доступен в SePlatform.HMI 2.0.

1.3.1

Исправления

- Ширина контекстного меню теперь лучше подстраивается под длину текста в элементах меню.
- При использовании [2.2.9.3. Диалог ввода учетных данных \(стр. 97\)](#) для подключения к серверу, не требующему пароля, в журнале сообщений больше не возникает ненужное предупреждение "Неизвестный ответ сервера на попытку авторизации (возможно, сервер не требует авторизации)".
- Исправлена ошибка, из-за которой в дереве сигналов [2.1.14. Дерево сигналов \(стр. 63\)](#) не работало перетаскивание сигнала или двойной клик по нему, если в названии сигнала присутствовали кавычки.
- Устранена причина, по которой в ОС Linux перемещение по элементам контекстного меню [2.2.1. Меню \(стр. 70\)](#) с помощью клавиатуры сбивалось при использовании клавиши "Влево".

Изменения документации

Редакция 1

В справочном руководстве:

- Описан новый компонент - окно для ввода учетных данных при подключении к серверу [2.2.9.3. Диалог ввода учетных данных \(стр. 97\)](#) и его свойства. Также обновлено описание и скриншоты окна подключения к серверу - [2.2.6. Диалог подключения к серверу \(стр. 80\)](#).
- Для элемента меню [2.1.10.1. Элемент меню \(стр. 41\)](#) описано новое свойство - KeyGesture ([стр. 42](#)). Свойство позволяет добавить в правую часть элемента меню текстовое описание сочетания "горячих" клавиш. Также обновлено описание свойства Items ([стр. 70](#)) компонента [2.2.1. Меню \(стр. 70\)](#) - здесь текст для правой части элемента меню указывается в JSON-парамetre `keyGesture`.
- Описано новое свойство компонента [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) - **Отображать кнопку Параметры (EditSettingsAllowed)**.
- Для компонента [2.1.14. Дерево сигналов \(стр. 63\)](#) пример использования функции **GetCurrentItemInfo()** переписан на язык JavaScript. Так как функция возвращает JSON-строку, то для ее обработки требуется разбор.

Редакция 2

Внутренние изменения. Содержимое документа не изменилось.

1.2

Новые возможности

- Разработаны новые компоненты:
 - **IntervalEditor** ([стр. 35](#)) - редактор интервала времени;
 - **ConnectDialog** ([стр. 80](#)) - диалог подключения к серверу;
 - **EventArgs** - буфер аргументов событий.
- Реализованы новые функции компонента **Converter** ([стр. 16](#)):
 - **Convert()**, конвертирующая значение внутри типа variant в значение указанного типа;
 - **VariantToLocalizedString()**, преобразовывающая значение типа variant в строку с учетом региональных настроек.

Улучшение

Иконка индикатора подключения к серверу **ConnectionIndicator** ([стр. 52](#)) теперь меняется при подключении к резервному серверу.

Исправленная ошибка

В окне календаря ([стр. 73](#)) даты больше не сбиваются при переключении месяца.

1.2.2

Новая возможность

Разработан вспомогательный тип [2.1.5. Информация о продукте \(стр. 24\)](#).

Улучшение

В редакторе [2.1.9.1. Редактор целого числа \(стр. 32\)](#) реализовано свойство **Localized**. В свойстве можно указать, должны ли числа в редакторе отображаться с учетом знака разделителя групп разрядов, указанного в региональных настройках.

Исправления

- В окне ввода значений [2.2.4. Диалог ввода значения \(стр. 77\)](#) числа, содержащие единицу ("1"), отображались некорректно.
- В окне подключения к серверу [2.2.6. Диалог подключения к серверу \(стр. 80\)](#) номер порта теперь отображается корректно - без учета знака разделителя групп разрядов, указанного в региональных настройках.

Изменения документации

Редакция 1

В справочном руководстве:

- описаны новые компоненты **IntervalEditor** ([стр. 35](#)) и **ConnectDialog** ([стр. 80](#));
- описаны новые функции компонента **Converter** ([стр. 16](#)): **Convert()** и **VariantToLocalizedString()**.

Во всем документе актуализированы скриншоты.

Редакция 2

В разделе [1.2. Подготовка к работе \(стр. 6\)](#) исправлены команды удаления расширения на ОС Linux.

В справочном руководстве описаны:

- новый вспомогательный тип [2.1.5. Информация о продукте \(стр. 24\)](#);
- новое свойство компонента [2.1.9.1. Редактор целого числа \(стр. 32\)](#) - **Localized**;
- ранее не описанный компонент [2.1.6. Буфер аргументов события \(стр. 25\)](#). Для демонстрации использования компонента разработан и описан проект-пример.

1.1

1.1.1

Исправления

- Ранее в окнах [2.1.13.9. Список файлов \(стр. 60\)](#) [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) и [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#) отображались файлы только тех типов, которые были указаны в свойстве **FileTypeFilter (Фильтр типа файлов)**, а если значение не было указано, файлы не отображались вовсе. Теперь, если оставить значение свойства пустым, в указанных диалогах будут отображаться файлы всех типов.
- Исправлена ошибка, из-за которой при закрытии подменю клавишей "Влево" меню закрывалось полностью.
- Устранена причина, по которой при работе в SePlatform.HMI.WebViewer меню не закрывалось, если с ним не взаимодействовать.

Изменения документации

Редакция 2

Актуализировано описание свойства **FileTypeFilter (Фильтр типа файлов)** компонентов [2.1.13.9. Список файлов \(стр. 60\)](#) [2.2.8.1. Диалог открытия файла \(стр. 83\)](#) и [2.2.8.2. Диалог сохранения файла \(стр. 88\)](#).